

# Analytical enclosure of the set of solutions of the three-species multivariate curve resolution problem

Tomass Andersons<sup>a</sup>, Mathias Sawall<sup>a</sup>, Klaus Neymeyr<sup>a,b</sup>

<sup>a</sup>Universität Rostock, Institut für Mathematik, Ulmenstraße 69, 18057 Rostock, Germany

<sup>b</sup>Leibniz-Institut für Katalyse, Albert-Einstein-Straße 29a, 18059 Rostock

---

## Abstract

In 1985 Borgen and Kowalski published a geometry-based mathematical approach in order to determine the set of feasible solutions of the multivariate curve resolution (MCR) problem for chemical systems with three species. Twenty years later Rajkó and István devised an algorithm for the analytical derivation of the feasible regions. They showed that the precise boundary of the solution set is piecewise representable in terms of analytical expressions for the boundary curve.

This paper generalizes the approach for finding analytical boundary curves by means of duality arguments, provides the precise functional form of the curves in detail, shows how to determine the contact change values and suggests improved analytical expressions which can numerically be evaluated in a stable way. Additionally, it offers detailed instructions for an algorithmic solution and provides the underlying analysis. A program code is presented which generates a piecewise functional representation of the boundary curve.

*Key words:* Multivariate curve resolution, nonnegative matrix factorization, pure component decomposition, Borgen-Rajkó plots, analytical derivation of the set of feasible solutions

---

## 1. Introduction

The multivariate curve resolution (MCR) problem in chemometrics consists of factorizing a component-wise nonnegative data matrix  $D \in \mathbb{R}_+^{k \times n}$  of spectral data according to the bilinear Lambert-Beer law as

$$D = CS^T. \quad (1)$$

If, for instance,  $D$  contains row-wise a sequence of spectra taken from an on-going chemical reaction with  $s$  chemical species, then (ideally) the columns of  $C \in \mathbb{R}_+^{k \times s}$  are the  $s$  time concentration profiles of the  $s$  chemical species. Further, the columns of  $S \in \mathbb{R}_+^{n \times s}$  are the associated pure component spectra. Typically, the factorization problem (1) has many solutions, namely component-wise nonnegative matrices  $C$  and  $S$  of the respective dimensions so that  $CS^T$  reconstructs  $D$ . The chemically true factors are among these solutions. This fact is known under the keyword of rotational ambiguity [17, 16, 1]. The sets of all feasible nonnegative factors  $C$  and  $S$  can be represented with respect to a low-dimensional basis of left- and right singular vectors of  $D$  by the Set of Feasible Solutions (SFS), see among others [6, 24, 25, 9, 26, 20]. For systems with three chemical species (with  $s = 3$ ) the SFS is a bounded two-dimensional set. For the basic and important case of three-component systems Borgen and Kowalski [6] published in 1985 a geometric algorithm to construct the SFS. The construction of the SFS is explained in Section 2 and this geometric representation is also known as Borgen plots or Borgen-Rajkó plots in literature. A generalization for noisy, experimental data has been suggested in [12]. Various other geometrical and numerical approaches have already been developed, see the survey [9] and [27, 13, 26] for some specific techniques. In 2005 Rajkó and István [24] published analytical expressions for the inner boundary curve (IBC) of the SFS also known as the limiting curve and laid the foundations for an analytical solution of the three-component MCR problem. An improvement of this method was provided by Beyramysoltan et. al. in 2014 [5]. Most of the above-mentioned publications underlies the important case of chemical systems with three species. We also make this restriction for the present work and assume that  $\text{rank}(D) = \text{rank}(C) = \text{rank}(S) = 3$ . This means for a singular value decomposition of the spectral data matrix  $D$  that three singular values are non-zero. Then the associated three left singular vectors of  $D$  are a basis of the column vectors of  $C$  and the associated right singular vectors also form a basis for representing the column vectors of the matrix factor  $S$ .

### 1.1. Data sets

The analytical studies in this work are accompanied and illustrated by evaluations for model problems. For the purpose of their comprehensibility the chosen dimensions are relatively small. A nonnegative 9-by-8 model matrix  $D$

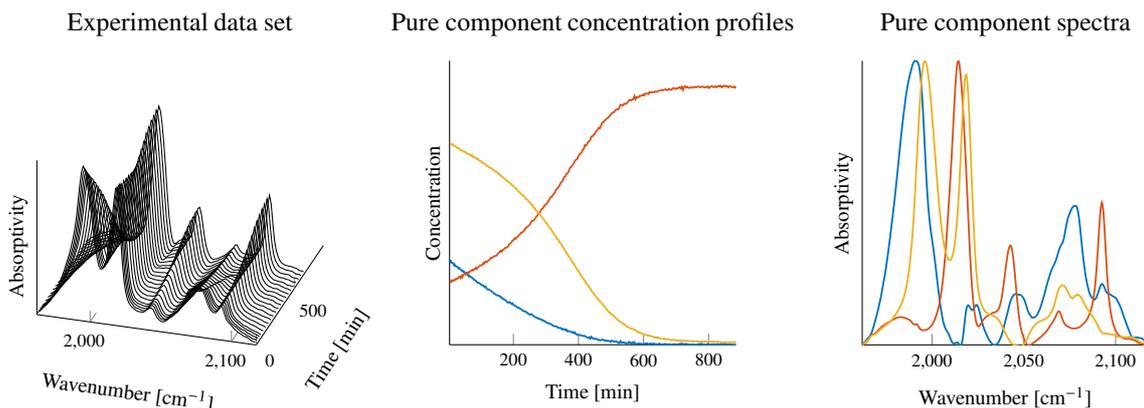


Figure 1: The data matrix and the true pure components of the FT-IR hydroformylation data set.

is introduced in Appendix A. Its small dimensions ensure that the construction steps of the boundary curve elements of SFS can easily be studied. We remark that this example matrix has no background of a spectroscopic data set; however, this fact does not imply any restrictions on the generality of our approach.

Moreover, to illustrate practical chemical applications, we apply our approach to an experimental FT-IR spectroscopic data set of the catalyst formation within the rhodium-catalyzed hydroformylation process, for more details see [14]. This reaction subsystem consists of three main absorbing components - the olefin species, the acyl complex and the hydrido complex. The aldehyde product of this reaction is not seen in the spectral window. The spectral data matrix  $D$  is a  $850 \times 645$  matrix, see Figure 1. The associated time interval for the 850 spectra is  $[4.7, 883.6]$ min and the 645 spectral channels are given in the wavenumber interval  $[1962.3, 2117.6]$  $\text{cm}^{-1}$ . The "true" pure component spectra and concentration profiles are obtained with the help of kinetic hard modelling and are also shown in Figure 1. The results are presented in Chapter 5.1.

### 1.2. Organization of the paper

First, Section 2 introduces the geometric background of the nonnegative matrix factorization problem underlying multivariate curve resolution and introduces the set of feasible solutions. The third section presents a geometric approach to construct the inner boundary curve of the set of feasible solutions together with closed-form expressions of the coordinates of the boundary curve. So-called contact change values which are the key quantities for an algorithmic implementation of an IBC computation algorithm are discussed in Section 4. Finally, Section 5 contains a complexity analysis and proves further properties of the IBC. Complicated mathematical formula, two proofs and the pseudocode of the algorithms are placed in the appendices.

### 1.3. Notation

Component-wise nonnegative matrices and vectors are simply called nonnegative and are represented for example by  $A \in \mathbb{R}_+^{c \times d}$ . The colon notation [10] is used to specify columns or rows of a matrix. For instance,  $A_{i,:}$  designates the  $i$ th row vector of  $A$  and  $A_{:,j}$  is the  $j$ th column vector of  $A$ .

List of variables:

$D$	(spectral mixture) data matrix,
$C$	first factor (columnwise concentration profiles),
$S$	second factor (columnwise spectral profiles),
$k$	number of spectra,
$n$	number of spectral channels,
$s$	number of chemical species, here $s = 3$ ,
$U, \Sigma, V^T$	factors of the singular value decomposition of $D$ ,
$\hat{T}$	non-scaled transformation matrix,
$T$	transformation matrix according to first singular vector scaling,
$\mathcal{M}/\mathcal{M}_d$	the primal/dual SFS,
$\mathcal{M}_C/\mathcal{M}_S$	the SFS of factor $C$ / factor $S$ ,
$\mathcal{F}/\mathcal{F}_d$	the primal/dual outer polygon (FIRPOL),
$\mathcal{F}_C/\mathcal{F}_S$	the outer polygon (FIRPOL) of factor $C$ / factor $S$ ,
$\mathcal{I}/\mathcal{I}_d$	the primal/dual inner polygon (INNPOL),

$\mathcal{I}_C/\mathcal{I}_S$	the inner polygon (INNPOL) of factor $C$ / factor $S$ ,
$\mathcal{T}/\mathcal{T}_d$	the primal/dual Borgen-2,3-triangle,
$\mathcal{B}$	the IBC (of the primal factor),
$\alpha$	rotation angle underlying the boundary curve formula and the IBC algorithm,
$\bar{t}, \bar{q}_r, \bar{q}_l/\dot{t}, \dot{q}_r, \dot{q}_l$	the edges of $\mathcal{T}$ / the vertices of $\mathcal{T}_d$ ,
$\bar{p}, \bar{s}_r, \bar{s}_l/\bar{p}, \bar{s}_r, \bar{s}_l$	the vertices of $\mathcal{T}$ / the edges of $\mathcal{T}_d$ ,
$t, q_r, q_l/p, s_r, s_l$	parameters for the edges of $\mathcal{T}$ / $\mathcal{T}_d$ in Hesse form and coordinates of the vertices of $\mathcal{T}_d$ / $\mathcal{T}$ ,
$\dot{p}$	the free vertex of the primal Borgen-2,3-triangle,
$p$	the analytical expression for IBC,
$\dot{v}, \dot{o}_l, \dot{o}_r/\bar{u}_l, \bar{u}_r$	vertices of $\mathcal{I}$ / edges of $\mathcal{F}$ for the construction of $\mathcal{T}$ ,
$\bar{v}, \bar{o}_l, \bar{o}_r/\dot{u}_l, \dot{u}_r$	edges of $\mathcal{F}_d$ / vertices of $\mathcal{I}_d$ for the construction of $\mathcal{T}_d$ ,
$v, u_l, u_r, o_l, o_r$	parameter pairs for the analytical expression of IBC,
$c_{x0}, c_{x1}, c_{x2}, c_{y0}, c_{y1}, c_{y2}, c_{d0}, c_{d1}, c_{d2}$	coefficients of the analytical expression for IBC,
$\alpha^1/\alpha^2/\alpha^3$	contact change value of the first/second/third kind,
$\alpha^{2l}/\alpha^{2r}/\alpha^{3l}/\alpha^{3r}$	left/right contact change value of the second/third kind,
$\alpha_i$	the $i$ -th contact change value,
$p^{[l]}$	analytical IBC on the $i$ -th interval with the $i$ -th set of parameters,
$m_{\mathcal{F}}/m_{\mathcal{I}}$	number of vertices of $\mathcal{F}$ / $\mathcal{I}$ ,
$m_a/m_b$	number of half-planes that form $\mathcal{F}$ / number of points whose convex hull results in $\mathcal{I}$ ,
$m_c$	number of contact change values,
$m$	the largest of the following values: $m_{\mathcal{F}}, m_{\mathcal{I}}$ ,
$d, e, g, h$	further auxiliary variables.

## 2. Geometric representation of the solution set

### 2.1. Inner polygon, outer polygon and the SFS

A standard approach to solving the MCR problem is based on singular value decompositions (SVD) [6, 20] for dimensionality reduction. Let  $D = U\Sigma V^T$  be a truncated singular value decomposition of the  $k \times n$  matrix  $D$  with the rank 3. Then the three nonzero singular values form the diagonal of the diagonal matrix  $\Sigma \in \mathbb{R}^{3 \times 3}$ . The columns of  $U \in \mathbb{R}^{k \times 3}$  and  $V \in \mathbb{R}^{n \times 3}$  are the associated left and right singular vectors. An SVD is not unique even if the singular values are pairwise different. Then there is still an orientation ambiguity, namely that pairs of left and right singular vectors  $(U_{:,i}, V_{:,i})$  can be substituted by  $(-U_{:,i}, -V_{:,i})$  without changing the properties of an SVD. Within the SVD plane these substitutions correspond to an axis flip. To ensure that we use fixed axes orientations for the following analysis, we assume that the SVD satisfies

$$\max(V_{:,i}) \geq \max(-V_{:,i}), \quad i = 1, 2, 3.$$

In words, the entry with absolute maximum in each column of  $V$  is chosen to be positive. This orientation is used in FACPAC [27]. However it is possible to use different sign conventions depending on the application, for example, the SignFlip Function as suggested by Bro, Acar and Kolda in [7], which aims to provide stable orientations for series of slightly altered data matrices. It has to be noted that the orientation has no impact on the resulting nonnegative factorizations.

Any nonnegative factorization  $D = CS^T$  can be represented with respect to the bases of left and right singular vectors [21]. This means that a transformation  $\hat{T} \in \mathbb{R}^{3 \times 3}$  exists so that

$$D = \underbrace{U\Sigma}_{=C} (\hat{T}^{-1} \hat{T}) \underbrace{V^T}_{=S^T}.$$

The rows of  $\hat{T}$  can be scaled according to

$$T_{i,:} = \frac{1}{\hat{T}_{i,1}} \hat{T}_{i,:}, \quad i = 1, 2, 3, \quad (2)$$

which further reduces some degree of freedom of the set of solutions of the MCR problem. Each row of the transformation is element-wise multiplied with the reciprocal of its first element. Thus the first column  $T_{:,1}$  is an all-ones

vector. This scaling is only permitted under the weak assumption of irreducibility of  $D^T D$ , see for example Theorem 2.2 in [30], which ensures that the first column  $\hat{T}_{:,1}$  contains no zeros. This mathematical property is a consequence of the Perron-Frobenius theory of nonnegative matrices. This row-scaling can be called the first singular vector scaling. Rajkó and István in [24] use a different scaling based on row sums of  $D$ . A one-one nonlinear transformation between these two scaling approaches exists showing their comparability [12]. This is closely related to Borgen norms, see [23].

Usually many matrices  $T$  exist which result in nonnegative factors  $C$  and  $S$ . This refers to the so-called factor ambiguity or rotational ambiguity underlying the set of solutions of the MCR problem, see [1] for an overview. For MCR problems with three chemical species, the set of possible nonnegative factors  $S$ , which can be called an SFS, can be represented by bounded sets in the two-dimensional real space. The same properties hold for the factor  $C$ . A formal definition of the SFS reads as follows.

**Definition 1.** *The SFS (Set of Feasible Solutions) for the factor  $S$  is defined as*

$$\mathcal{M}_S = \left\{ x \in \mathbb{R}^{2 \times 1} : \exists T \in \mathbb{R}^{3 \times 3}, T_{1,2:3} = x^T, T_{:,1} = \mathbf{1}, \text{rank}(T) = 3, C = U\Sigma T^{-1} \geq 0, S^T = TV^T \geq 0 \right\} \quad (3)$$

with the three-dimensional all-ones column vector  $\mathbf{1}$ . Analogously, the SFS  $\mathcal{M}_C$  for the other (dual) factor  $C$  can be determined by a similar construction based on  $T^{-1}$ .

Note that the existence of a regular  $T$  in Definition 1 implies the correctness of the equation  $D = CS^T$ , because

$$D = U\Sigma V^T = U\Sigma(T^{-1}T)V^T = (U\Sigma T^{-1})(TV^T) = CS^T.$$

The inherent structure of the nonnegative matrix factorization problem imposes various mathematical and geometric restrictions on the set of feasible solutions. Geometric construction approaches and many numerical approximation algorithms have been suggested to construct or to compute the SFS. See the references [15, 19, 6, 24, 12] and many others. A geometrically constructed SFS for noise-free, ideal data goes back to the influential work by Borgen and Kowalski [6]. A generalization to noisy, experimental data has been suggested in [12, 28].

Key objects for the construction of the SFS are an outer polygon  $\mathcal{F}$ , where the letter F refers to the outdated name FIRPOL, and an inner polygon  $\mathcal{I}$  which stands for INNPOL. These polygons exist for the factor  $C$  and also for  $S$ . The respective target can be expressed by an index, that is we consider  $\mathcal{I}_C$  and  $\mathcal{F}_C$  for the factor  $C$  and  $\mathcal{I}_S$  and  $\mathcal{F}_S$  for the factor  $S$ . The precise definitions of these polygons read as follows.

**Definition 2.** *The polygon  $\mathcal{F}$  is the intersection of affine half-planes*

$$\mathcal{F} = \left\{ x \in \mathbb{R}^2 : x^T a_i \geq -1, \quad i = 1, \dots, m_a \right\}$$

and  $\mathcal{I}$  is defined as

$$\mathcal{I} = \text{convhull}(\{b_j \in \mathbb{R}^2 : j = 1, \dots, m_b\})$$

where for the factor  $S$  the  $a_i$  and  $b_j$  are given by

$$a_i = \frac{(V^T)_{2:3,i}}{(V^T)_{1,i}}, \quad m_a = n, \quad b_j = \frac{((U\Sigma)_{j,2:3})^T}{(U\Sigma)_{j,1}}, \quad m_b = k$$

and similarly for the factor  $C$

$$a_i = \frac{((U\Sigma)_{i,2:3})^T}{(U\Sigma)_{i,1}}, \quad m_a = k, \quad b_j = \frac{(V^T)_{2:3,j}}{(V^T)_{1,j}}, \quad m_b = n.$$

The outer polygon  $\mathcal{F}$  is a superset of  $\mathcal{M}$ . This means that  $\mathcal{F}_C \supset \mathcal{M}_C$  and  $\mathcal{F}_S \supset \mathcal{M}_S$ . Further,  $\mathcal{F}$  encloses the inner polygon  $\mathcal{I}$ , see for example [20]. See Section 2.3 for more details on the geometric construction and in which way the inner polygon serves as an ‘‘interior’’ restriction for the SFS.

## 2.2. Duality principles in MCR

Definition 2 strongly motivates the concept of duality, as  $\mathcal{F}_C$  and  $\mathcal{I}_S$  as well as  $\mathcal{F}_S$  and  $\mathcal{I}_C$  are generated by the same vectors. The duality theory for these pairs of inner and outer polygons has its roots in [11, 22] and has also been explored in [26]. The basic relations of duality are briefly presented below.

**Definition 3.** A nonzero (point)  $a \in \mathbb{R}^2$  is dual to the affine line

$$\{x \in \mathbb{R}^2 : x^T a = -1\}. \quad (4)$$

**Remark 1.** The line construction (4) is closely related to the Hesse normal form of a line. From  $x^T a = -1$  we get the equivalent condition in the shape of the Hesse normal form for a nonzero vector  $a \in \mathbb{R}^2$

$$x^T \frac{a}{\|a\|} - \left(-\frac{1}{\|a\|}\right) = 0.$$

This describes an affine straight line oriented orthogonal to the vector  $a$  whose smallest distance  $1/\|a\|$  to the origin is attained in the point  $-a/\|a\|^2$ .

Duality induces various relations between the inner and outer polygon and the SFS for the two factors  $C$  and  $S$ . If we want to refer to the so-called primal factor, then we use the notation  $\mathcal{F}, \mathcal{I}, \mathcal{M}$  and add a subscript  $d$  for the dual factor, namely  $\mathcal{F}_d, \mathcal{I}_d, \mathcal{M}_d$ . In the following examples,  $S$  is chosen as the primal factor, unless stated otherwise. Definition 3 enables useful relationships between the two (dual) geometric representations. Most notably, a vertex of  $\mathcal{I}$  is dual to a line that contains a facet of  $\mathcal{F}_d$  and a line that contains a facet of  $\mathcal{I}$  is dual to a vertex of  $\mathcal{F}_d$ . Some duality relations are illustrated in Figure 2. For the mathematical proofs of these relations see [26, 3]; a deepened analysis is planned for a forthcoming publication. These relations allow us to continue the theoretical and practical considerations without having to specify which of the two factors is the primal one.

With the help of duality the inner and outer polygon can be determined easily. This as well as the convex hull algorithm from MATLAB [18] was used in the improved method of Beyramysoltan et. al. [5]. In our method we assume that the inner and outer polygons have already been calculated.

### 2.3. Geometric construction of the set of feasible solutions

The polygons  $\mathcal{F}$  and  $\mathcal{I}$  are the key-geometric objects for the construction of the set of feasible solutions. The construction approach was introduced by Borgen and Kowalski [6] with their simplex rotation algorithm. According to (2) and (3) a factorization of  $D$  corresponds to a triangle with the three vertices  $T_{1,2,3}$ ,  $T_{2,2,3}$  and  $T_{3,2,3}$ . Such a factorization  $D = CS^T$  is feasible, that is it yields nonnegative factors  $C$  and  $S$ , if and only if the triangle includes the inner polygon  $\mathcal{I}$  and is enclosed by the outer polygon  $\mathcal{F}$ ; for these fundamental relations see for example Theorem 3.11 in [12]. Each vertex of the triangle represents a possible spectral profile or a possible concentration profile of a pure component. Consequently, the nonnegative matrix factorization problem is equivalent to the geometric problem of determining a nested triangle with the mentioned properties. The SFS is the set of all vertices of all feasible triangles and expresses the factor ambiguity of the nonnegative matrix factorization problem. According to [6] the geometric construction of the set of all factorizations can be achieved by the construction of boundary cases. These boundary cases are explained next.

A *Borgen- $a, b$ -triangle* is defined as a triangle that lies between the boundaries of the inner polygon  $\mathcal{I}$  and the outer polygon  $\mathcal{F}$ , that has a vertices located on the boundary of  $\mathcal{F}$ , and  $b$  of its edges touch  $\mathcal{I}$ . It is a generalization of the Borgen triangle, as given in [24], [5] or the limiting simplex in [6]. Such triangles approximate the SFS and are called *Borgen-3, 3-triangles* according to our definition.

Furthermore, each feasible matrix  $T$ , according to Definition 1, results in a Borgen- $a, b$ -triangle. If the triangle does not touch neither  $\mathcal{F}$  nor  $\mathcal{I}$ , then it is a *Borgen-0, 0-triangle*.

Main focus for the construction of the the SFS is on Borgen-2, 3-triangles  $\mathcal{T}$ . They consist of two outermost vertices with three innermost edges, that is they have two vertices located on the boundary of  $\mathcal{F}$ , and all their edges touch the inner polygon  $\mathcal{I}$ . Without rotation this third vertex is as close as possible to the inner polygon and represents a boundary of the SFS through the limitations imposed by the positions of  $\mathcal{F}$  and  $\mathcal{I}$ , and we call it a *free vertex*. Such innermost boundary points of the set of feasible solutions are the free vertices. A typical triangle satisfying these conditions is illustrated for the primal factor in Figure 3. The primal triangle is plotted green with vertices plotted in cyan. Figure 3 also shows the dual lines (in cyan) which form the dual triangle  $\mathcal{T}_d$ . Note that the dual triangle of a Borgen-2, 3-triangle is always a Borgen-3, 2-triangle.

By rotating the Borgen-2, 3-triangle around the inner polygon  $\mathcal{I}$  a continuous curve of innermost boundary points is constructed. During the rotation process the two outermost vertices are continuously moved to the boundary of the outer polygon and the innermost vertex is always constructed according to the rules given above. The curve is given by the positions of all free vertices which have been constructed during the triangle rotation process. Parts of this curve form the inner boundary of the SFS whereas the outer boundary of the SFS is determined by the boundary of the outer polygon. Hence we call the constructed curve the inner boundary curve (IBC) of the given three-component problem and refer to it as  $\mathcal{B}$ . If we intersect the IBC with the boundary of the outer polygon  $\mathcal{F}$ , then the enclosed finite regions which are located in the outer polygon define the SFS. These areas are plotted gray in Figure 3. In this

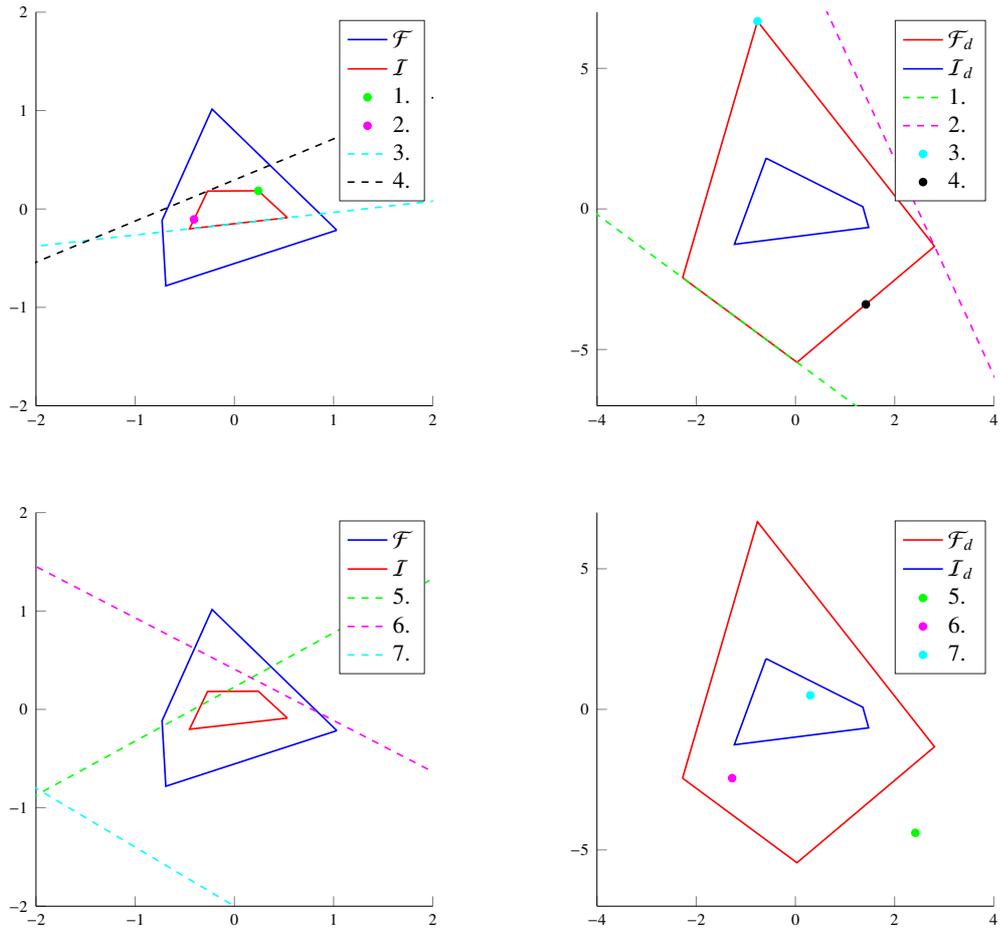


Figure 2: Seven duality relations illustrated for the three-component model problem. The numbers represent different cases of duality. The primal polygons are shown on the left. The dual objects are plotted on the right. The same color is used for dual pairs of points and affine straight lines. Vertices of one polygon correspond to lines that contain edges of the dual polygon. The relations between  $\mathcal{F}$  and  $\mathcal{I}_d$  as well as  $\mathcal{I}$  and  $\mathcal{F}_d$  are reflected in the chosen color scheme. The lower two plots illustrate these duality relations for points that are not on the boundary of the respective polygons.

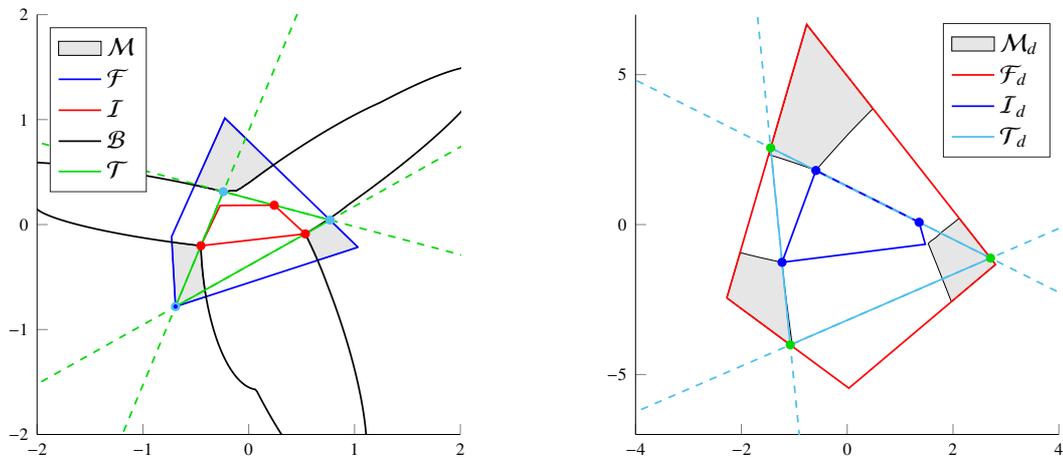


Figure 3: Duality of a Borgen-2,3-triangle  $\mathcal{T}$  whose vertices are plotted in cyan (left plot) for the model problem as given in Appendix A. The dual triangle  $\mathcal{T}_d$  (a Borgen-3,2-triangle) is shown in the right plot and is drawn with edges in cyan. The free vertex is a point on the inner boundary curve (IBC)  $\mathcal{B}$ .

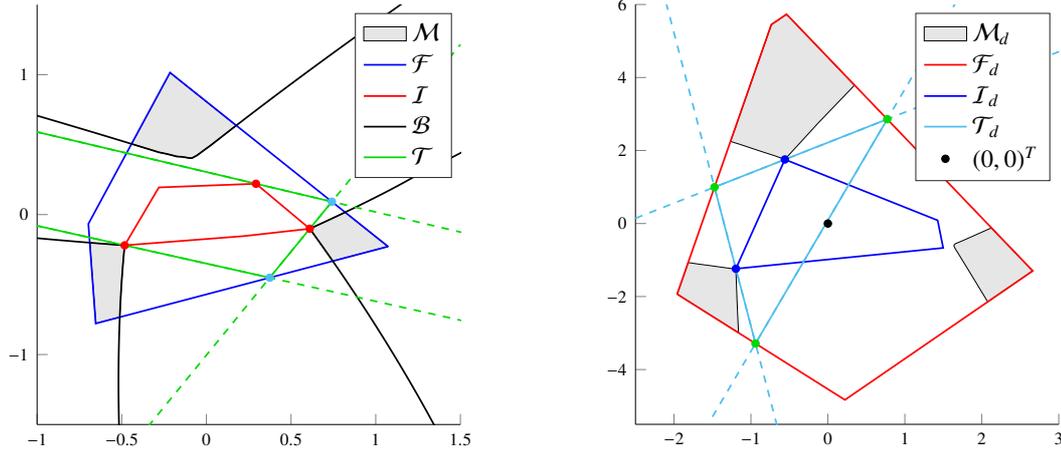


Figure 4: We modify the model problem in order to construct a rare limit case and consider the IBC  $\mathcal{B}$  for the model matrix (13) after subtraction of 9 from each of its matrix elements. Left: A Borgen-2, 3-triangle  $\mathcal{T}$  does not always exist (see green lines) for the primal problem. Right: The corresponding dual triangle  $\mathcal{T}_d$  violates the necessary condition of enclosing the origin. Then the IBC  $\mathcal{B}$  is not closed, but tends to infinity.

figure, the IBC is plotted by a black line. Its parts in the outer polygon are of importance for the SFS construction. However, the IBC curve also exists outside the polygon  $\mathcal{F}$ , see Figure 3.

In many cases the IBC is a continuous (closed) curve. However, in some situations the IBC can tend to infinity which prevents the existence of a continuous or closed IBC. An example situation is shown in Figure 4. A singularity of the IBC occurs, if the dual triangle  $\mathcal{T}_d$  does not include the origin for certain rotation angles. The SFS is not affected by such singularities of the IBC since this occurs outside the outer polygon where the feasibility of triangles is not satisfied.

### 3. Function representation of the IBC

In 2005 Rajkó and István [24] published analytical expressions for a piecewise representation of the IBC. Here we deepen this analysis, introduce a general approach of presenting the IBC by piecewise analytical expressions in function form on the basis of contact change values, generalize the classical approach by using duality arguments and derive analytical expressions with a focus on their numerical stability. The goal is to present a simple, efficient and stable algorithmic implementation by using duality relations.

#### 3.1. Derivation of a closed-form expression for points on the IBC

The Borgen-2, 3-triangles construction only requires the coordinates of the vertices of  $\mathcal{I}$  and also the edges of  $\mathcal{F}$ . The location of the free vertex depends on these given parameters, as well as on the triangle rotation. See Figure 5 for the variable names of the Borgen-2, 3-triangle  $\mathcal{T}$  and its dual Borgen-3, 2-triangle  $\mathcal{T}_d$ . All points are denoted by lowercase letters with a dot above. The corresponding dual lines are written with a bar above the same lowercase letter. The edges of the polygons are also represented by their corresponding line equations and no further distinction between lines and edges is required. The subscript letters  $l$  and  $r$  stand for left and right with the view direction from the point  $\dot{i}$  of the dual plot.

Rajkó and István [24] and Beyramysoltan et. al. [5] use coordinate pairs for the points  $\dot{e} = (x, y)$  and slope-intercept form for lines

$$\bar{e} = \{(x, y) : y = g + hx\}.$$

We prefer to write line equations in the Hesse form, see (4),

$$\bar{e} = \{(x, y)^T \in \mathbb{R}^2 : e_1 x + e_2 y = -1\}, \quad \dot{e} = (e_1, e_2)^T \in \mathbb{R}^2 \setminus (0, 0)^T. \quad (5)$$

This offers a double advantage. The line  $\bar{e}$  is dual to the point  $\dot{e}$ ; compare this with Definition 3. Consequently, no computation is required for this transformation and no numerical precision is lost. Only lines passing through the origin cannot be of the form (5). However, duality does not allow lines through the origin. Thus (5) is the appropriate notation for our scope of application and increases the numerical stability.

The approach [24] considers a rotation of a triangle as a function of the  $x$ -coordinate of  $\dot{s}_l$ . Here we choose the angle  $\alpha$  between the positive  $x$ -axis and a vector towards  $\dot{i}$  in the dual plot as the free variable. This representation has



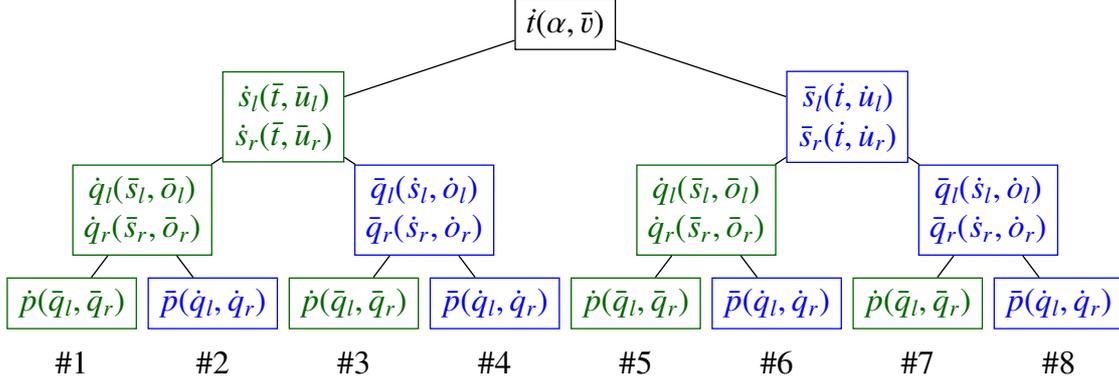


Figure 6: Possible derivation routes for the boundary curve by using duality relations underlying the nonnegative matrix factorization problem. Some of the steps refer to the primal triangle and others to the dual triangle. Green color is used if the coordinates of a point are calculated from two lines. Blue color indicates if a line is calculated from two points.

2. Solving the dual problem of determining a line  $\bar{e} = \{(x, y)^T \in \mathbb{R}^2 : e_1x + e_2y = -1\}$  that passes through two different points  $\dot{g} = (g_1, g_2)^T$  and  $\dot{h} = (h_1, h_2)^T$ . This yields the same linear system of equations as considered above with the solution

$$e_1 = \frac{g_2 - h_2}{g_1h_2 - g_2h_1}, \quad e_2 = \frac{g_1 - h_1}{g_1h_2 - g_2h_1}. \quad (10)$$

Numerical stability is a critical issue in the evaluation of the various mathematical terms derived here. In general, we present a set of expressions with the best numerical stability as found by testing various ways of computing the IBC curve. The steps for the construction of the IBC point  $\dot{p}$  can be accomplished via the primal triangle  $\mathcal{T}$  and/or the dual triangle  $\mathcal{T}_d$ , as the choice of line equation allows a simple transformation from one to another. All the possible construction routes for  $\dot{p} = (p_1, p_2)$  are illustrated in Figure 6. Route #3 in Figure 6 only performs computations within the primal plot to determine  $\dot{p}$ , and this route is similar to the approach of Rajkó and István [24] as well as Beyramysoltan et. al. [5]. A discussion on the best route for an algorithmic implementation follows below.

In dependence on the angle  $\alpha$  and the two-dimensional vector parameters  $v, s_l, s_r, o_l$  and  $o_r$  we get the following analytical expressions for  $\dot{p} = (p_1, p_2)^T$

$$\begin{aligned} p_1 &= \frac{c_{x0} + c_{x1} \tan(\alpha) + c_{x2} \tan^2(\alpha)}{c_{d0} + c_{d1} \tan(\alpha) + c_{d2} \tan^2(\alpha)}, \\ p_2 &= \frac{c_{y0} + c_{y1} \tan(\alpha) + c_{y2} \tan^2(\alpha)}{c_{d0} + c_{d1} \tan(\alpha) + c_{d2} \tan^2(\alpha)}, \end{aligned} \quad (11)$$

where complicated formula for the nine coefficients  $c_{x0}, c_{x1}, c_{x2}, c_{y0}, c_{y1}, c_{y2}, c_{d0}, c_{d1}$  and  $c_{d2}$  are given in Appendix B.

### 3.2. Contact change values

The IBC is the set of all constructable points  $\dot{p}$ . The true complexity of the Expression (11) lies in the fact that the parameters  $v, s_l, s_r, o_l$  and  $o_r$  change in dependence on the angle  $\alpha$ . The reason for this is that the contact conditions of the Borgen-2, 3-triangle changes with the rotation angle  $\alpha$  and so do the parameter values. This concept of parameter values underlying  $\dot{p}$  changing at certain values of  $\alpha$  is similar to the approach of contact change points by Aggarwal et. al. [2].

**Definition 4.** A value  $\alpha_i \in [-\pi, \pi)$  for  $i = 1, \dots, m_c$  is called the  $i$ -th contact change value if one of the vertices of the corresponding Borgen-2, 3-triangle coincides with a vertex of  $\mathcal{F}$  or if one of the edges of the Borgen-2, 3-triangle is collinear with an edge of  $\mathcal{I}$ . The corresponding point  $\dot{p}(\alpha)$  on the IBC is called the contact change point. If more than a single vertex or edge is responsible for the same contact change value, then they are still counted separately.

The contact change values  $\alpha_i$  partition the angle interval  $[-\pi, \pi)$  in subintervals

$$-\pi = \alpha_0 \leq \alpha_1 \leq \dots \leq \alpha_i \leq \alpha_{i+1} \leq \dots \leq \alpha_{m_c} \leq \alpha_{m_c+1} = \pi.$$

Each subinterval  $[\alpha_i, \alpha_{i+1})$  stands for a unique set of parameters used in the construction of the Borgen-2, 3-triangle. The parameters change at each contact change value. Closed-form expressions can be derived for  $\dot{p}$  in each interval. All this results in the IBC function for all angle values  $\alpha \in [-\pi, \pi)$ .

**Definition 5.** The IBC function is an interval-wise defined function with changing sets of parameters depending on the interval. The dependence on the particular interval is expressed by an additional superindex in brackets. Thus we consider

$$p : [-\pi, \pi) \rightarrow \mathbb{R}^2, \quad \alpha \mapsto \begin{cases} p^{[m_c]}(\alpha) & \text{for } \alpha \in [\alpha_0, \alpha_1), \\ p^{[1]}(\alpha) & \text{for } \alpha \in [\alpha_1, \alpha_2), \\ \vdots & \\ p^{[i]}(\alpha) & \text{for } \alpha \in [\alpha_i, \alpha_{i+1}), \\ \vdots & \\ p^{[m_c]}(\alpha) & \text{for } \alpha \in [\alpha_{m_c}, \alpha_{m_c+1}), \end{cases}$$

where  $p^{[i]}(\alpha)$  equals  $\dot{p}$  by (11) in the  $i$ -th interval  $\alpha \in [\alpha_i, \alpha_{i+1})$  with its specific (and in this interval constant) set of parameters.

The repetition of the function  $p^{[m_c]}(\alpha)$  in the first and the last subinterval of  $[-\pi, \pi)$  reflects the fact that  $\alpha = \pi$  is not necessarily a contact change value, but only a point in which the upper and lower semicircles meet.

The approach of Beyramysoltan et. al. [5] suggests to only calculate the IBC function for the intervals which result in the points  $\dot{p}$  that are located in  $\mathcal{F}$ , however this requires the search for all Borgen-3, 3-triangles. This method does not provide any significant improvement of the calculation time for our implementation.

#### 4. Algorithmic approach for constructing the boundary curve

The main complexity of the boundary curve construction consists in finding the contact change values  $\alpha_i$ . Interval-wise the associated parameter sets are fixed and they change from one to the next interval. This section deals with some overall considerations of an algorithmic solution of this problem, but is not an attempt at a complete generalization of all possible approaches. We explain an approach based on determining tangents of a polygon by using a convex hull construction. Alternative algorithms are possible, for instance those which are based on determining the intersection of an affine line with a general polygon. Here, we prefer to compute the contact change values by means of a convex hull algorithm. This provides the parameters for the construction route #1 shown in Figure 6.

Contact change values can be categorized into three main types and different steps are to be taken to find them all. The pseudo-code for the three types and both approaches can be found in Appendix C.

##### 4.1. Contact change values of the first kind $\alpha^1$ : Primal inner edge coincidence

The contact change values of the first kind represent the changing parameter values in the top level of Figure 6. This is a situation in which an edge of the primal triangle  $\mathcal{T}$  completely covers an edge of the inner polygon. This situation is dual to the geometric incidence of a vertex of the dual outer polygon with a vertex of the dual triangle  $\mathcal{T}_d$ . Hence, each vertex of the dual outer polygon  $\mathcal{F}_d$  corresponds to a specific contact change value of the first kind. Therefore, we find all contact change values  $\alpha^1$  by calculating the two-argument arctangent values (inverse tangent value) of each vertex of  $\mathcal{F}_d$ . By running through such contact change values of the first kind, the present parameter  $v$  is replaced by the next, directly following edge of  $\mathcal{F}_d$ . Figure 7 illustrates such a contact change value of the first kind  $\alpha^1$  in the primal and dual representation. Additionally, see Figure 8 for the construction. We call this situation a primal inner edge coincidence.

##### 4.2. Contact change values of the second kind $\alpha^2$ : Primal outer vertex coincidence

Contact change values of the second kind concern a primal outer vertex coincidence, which is explained next. Such contact change values correspond to changing parameter values in the second level (from the top) of Figure 6, that is,  $u_l$  or  $u_r$  changes its value or equivalently  $\bar{u}_l$  or  $\bar{u}_r$  in the dual plot. We calculate two tangents of  $\mathcal{I}$  for each vertex of  $\mathcal{F}$  by computing the convex hull of vertices of  $\mathcal{I}$  together with a single vertex of  $\mathcal{F}$ . Two edges of this convex hull are the desired tangents  $\bar{t}$ . The algorithmic pseudocode of these steps is listed in Appendix C and contains the details on the convex hull construction.

Without additional effort we transform the two tangents to the line  $\bar{t}$  and the dual point  $i$ . Then  $\alpha^2$  is the two-argument inverse tangent value of the point  $i$ . The change of parameters  $u_l$  or  $u_r$  for the inner boundary curve function is given by the following edge of  $\mathcal{F}$ . See Figure 9 for an illustration and Figure 10 for the construction.

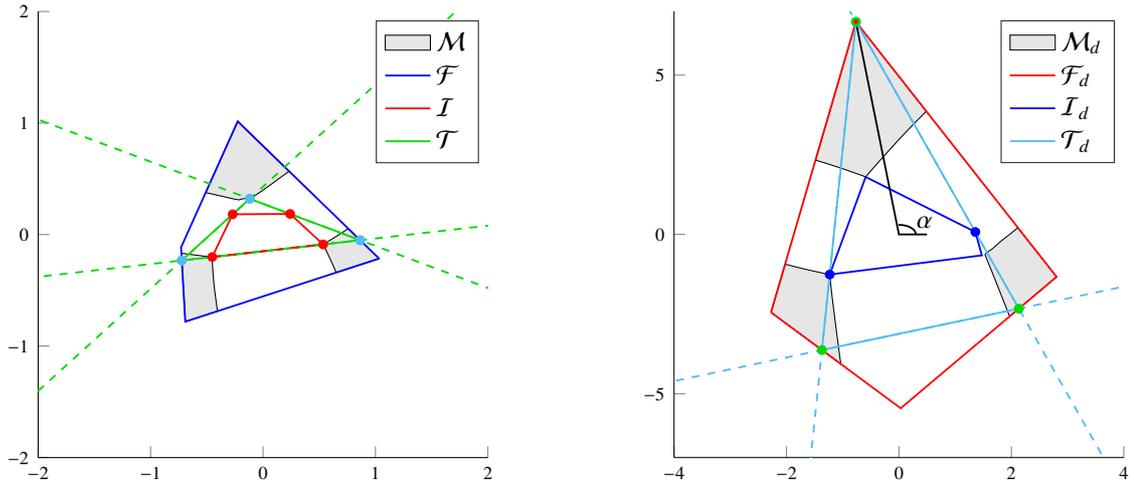


Figure 7: A contact change value of the first kind  $\alpha^1$  for the model problem is illustrated in the primal SFS (left) and the dual SFS (right). For such contact change values an edge  $\bar{i}$  of  $\mathcal{T}$  completely covers an edge of the primal triangle  $\mathcal{I}$ , see the lowermost green line in the left plot. In the dual plot this corresponds to a vertex  $i$  of  $\mathcal{T}_d$  (uppermost green circle in the right plot) which coincides with a vertex of  $\mathcal{F}_d$ .

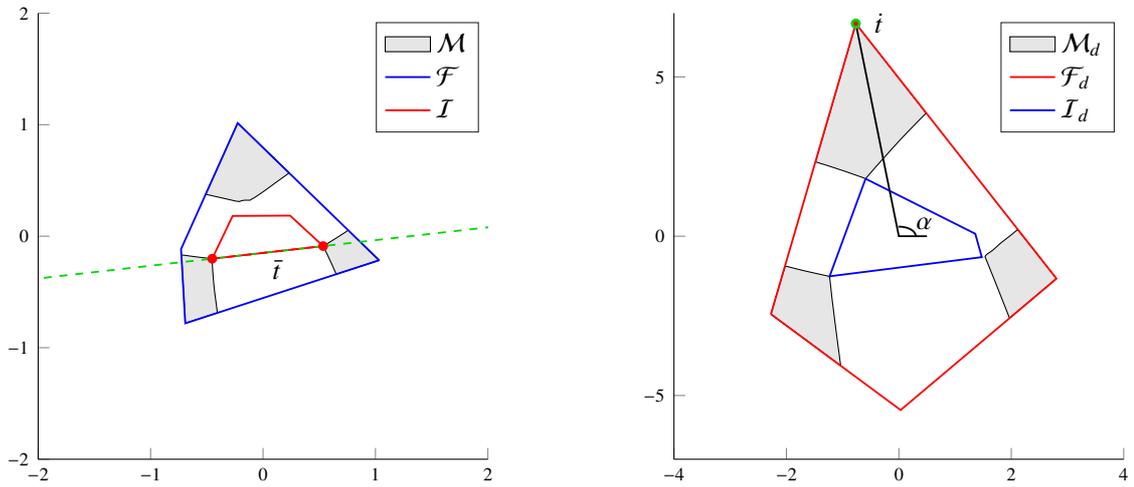


Figure 8: Calculation of a contact change value of the first kind  $\alpha^1$  from Figure 7 by computing the inverse tangent (arctan) value which corresponds to the (marked) vertex of the dual outer polygon (right subplot).

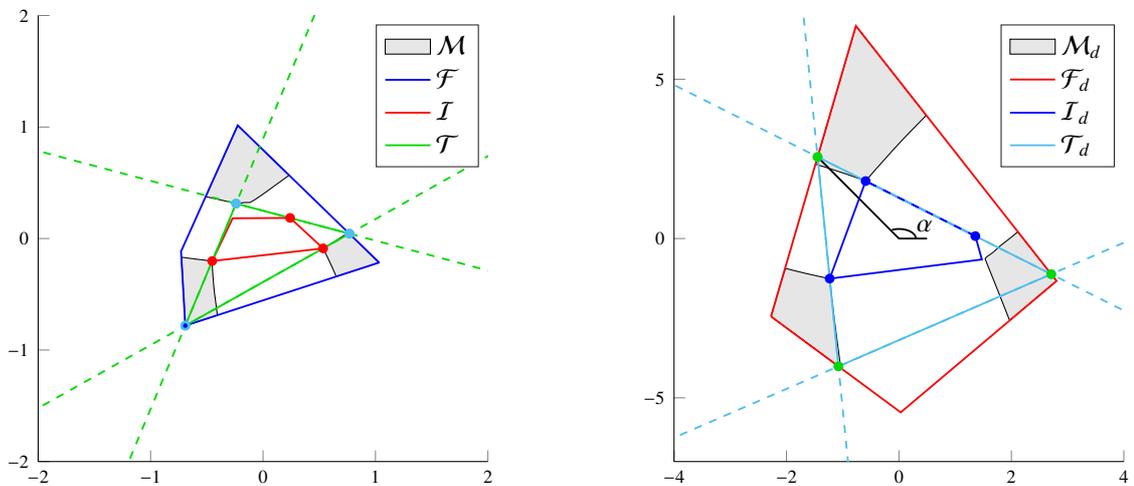


Figure 9: A left contact change value of the second kind  $\alpha^{2l}$  for the model problem. The vertex  $s_l$  of  $\mathcal{T}$  coincides with a vertex of  $\mathcal{F}$  and the edge  $\bar{s}_l$  of  $\mathcal{T}_d$  contains an edge of  $\mathcal{I}_d$ .

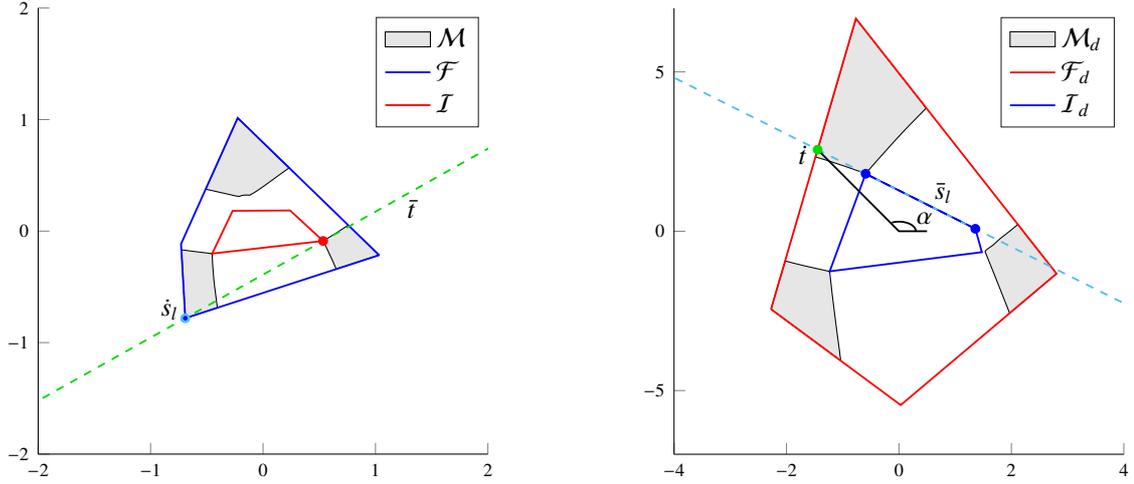


Figure 10: The calculation of the left contact change value of the second kind  $\alpha^{2l}$  from Figure 7.

#### 4.3. Contact change values of the third kind $\alpha^3$ : A second primal inner edge coincidence

The contact change values of the third kind represent the changing parameter values in the third level (from the top) of Figure 6, that is,  $o_l$  or  $o_r$  changes its value. So we distinguish such contact change values of the left and the right type. See Figure 5 on  $o_l$  or  $o_r$  and all further variables. The contact change values of the third kind refer to a geometric setup which is similar to that of the first kind. However, due to a different orientation of the inner edge coincidence an active edge  $o_l$  or  $o_r$  changes its value. Compare this with Section 4.1 where the edge  $v$  is changed.

We calculate the two tangents of  $\mathcal{I}_d$  for each vertex of  $\mathcal{F}_d$  with the convex hull algorithm. If the left contact change value of the third kind is obtained by constructing the dual triangle  $\mathcal{T}_d$ , then the respective vertex of  $\mathcal{F}_d$  represents the point  $\hat{q}_l$  and the tangent stands for the line  $\bar{s}_l$ .

Then we use the dual point  $\hat{s}_l$  of the line  $\bar{s}_l$ . The point  $\hat{s}_l$  lies on  $\mathcal{F}$  and it allows us to obtain the left and the right tangents of  $\mathcal{I}$  by means of the convex hull algorithm. One of the tangents is the line  $\bar{q}_l$  and can be discarded. The other tangent is  $\bar{i}$ . Then it can be transformed to the dual point  $i$  and its 2-argument inverse tangent is the desired contact change value  $\alpha^3$ . A right contact change value of the third kind can be obtained by following the same steps with the other tangent of the  $\mathcal{I}_d$ .

The change of either parameter  $o_l$  or  $o_r$  for the construction of the inner boundary curve function consists of substituting the present edge by the following edge of  $\mathcal{F}$ . See Figure 11 for an illustration of the geometric setup and Figure 12 for the construction.

#### 4.4. Computing the IBC function

Once we have determined all contact change values together with their corresponding changing parameters, they are sorted in ascending order. Each contact change value changes only a single pair of parameters whereas the remaining parameters can be adopted from the previous subinterval. This easily allows us to fill the parameter table for all subinterval as introduced in Definition 5. A subsequent evaluation of the analytical expressions (11) yields the IBC as a function of the rotation angle  $\alpha$  in each of the subintervals.

The IBC function can be evaluated with any desired precision. The precision is only limited by the computer arithmetic used. Hence the presented approach is a truly analytical approach. Further, the accuracy of the IBC curve can be adapted to any level of precision for regions of particular interest. However, in most practical situations the double precision (64 bit) arithmetic of a standard personal computer appears to be sufficient for most applications. In a final step the SFS is found as the intersection of the exterior of the IBC curve with the outer polygon  $\mathcal{F}$ .

## 5. Discussion and further result

Next, an application is presented for the three-component consecutive chemical reaction system as introduced in Section 1.1. Further, we provide some properties of the IBC functions and of its algorithmic evaluation. Our analysis also encompasses a complexity analysis of the algorithm and the discussion of some problematic cases.

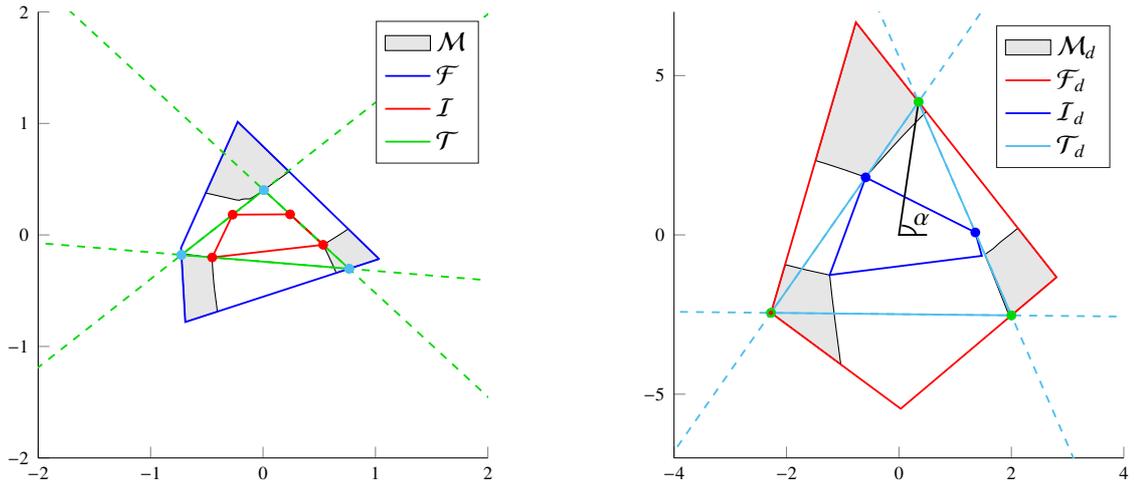


Figure 11: A right contact change value of the third kind  $\alpha^{3r}$  for the model problem. The edge  $\bar{q}_r$  of  $\mathcal{T}$  covers an edge of  $\mathcal{I}$  and the vertex  $\dot{q}_r$  of  $\mathcal{T}_d$  coincides with a vertex of  $\mathcal{F}_d$ .

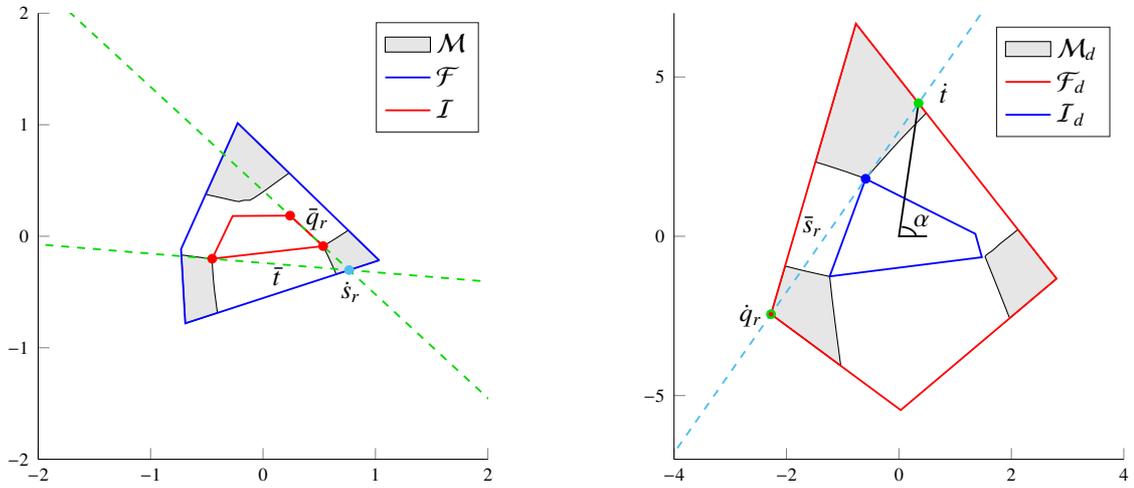


Figure 12: The calculation of the right contact change value of the third kind  $\alpha^{3r}$  for the geometric situation as shown in Figure 7.

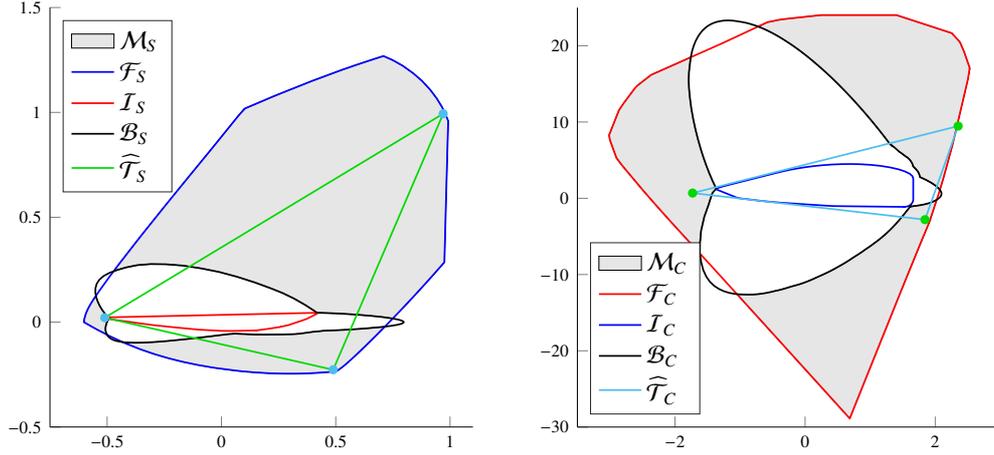


Figure 13: The IBC  $\mathcal{I}$ , the outer polygon  $\mathcal{F}$ , the inner polygon  $\mathcal{I}$ , the SFS  $\mathcal{M}$  for both factors of the experimental FT-IR spectroscopic data set as introduced in Section 1.1. Furthermore, the triangles that correspond to the true pure components in Figure 1 are shown and are denoted by  $\widehat{\mathcal{T}}_S$  and  $\widehat{\mathcal{T}}_C$  respectively.

### 5.1. Application to an experimental problem

The structure of the FT-IR data set is explained in 1.1 and the three concentration profiles and the associated spectra of the pure components are shown in Figure 1. Here the analysis with the IBC algorithm follows.  $\mathcal{F}_S$  is spanned by 217 points and  $\mathcal{F}_C$  - by 103 points. The outer boundaries required for our algorithm are calculated using the polygon inflation algorithm for noisy data, see [27, 30, 28] and then the inner boundaries are found with the help of duality. This step is also very similar to the use of duality in [5]. Then the IBC algorithm is applied and Figure 13 shows the resulting SFS for both factors in gray color.  $\mathcal{B}_S$  contains 743 subintervals and subfunction.  $\mathcal{B}_C$  contains 857 subintervals. The total calculation time for the IBC of both factors typically was less than 0.6 seconds in a MATLAB 2019b (single core) implementation running on a standard desktop computer (3.6GHz Intel CPU).

### 5.2. Number of contact change values

The number of contact change values equals the number of subfunctions of the piecewise-defined IBC function. Hence this number is a meaningful key indicator for the total computing time.

Let  $m_{\mathcal{I}}$  be the number of vertices of  $\mathcal{I}$  and  $m_{\mathcal{F}}$  be the number of vertices of  $\mathcal{F}$ . The number of the contact change values of the first kind is exactly  $m_{\mathcal{I}}$ . The number of the left contact change values of the second kind is  $m_{\mathcal{F}}$ . The same holds for the right contact change values of the second kind. The number of the left and also that of the right contact change values of the third kind equals  $m_{\mathcal{I}}$ . Therefore, the total number of contact change values  $m_c$  sums up to

$$m_c = 3m_{\mathcal{I}} + 2m_{\mathcal{F}} \leq 3m_b + 2m_a, \quad (12)$$

including possibly overlapping values. The constants  $m_a$  and  $m_b$  in Eq. (12) are introduced in Definition 2 and equal the dimensions  $k$  or  $n$  of the  $k \times n$  spectral mixture data matrix  $D$ . If the SFS for the spectral factor  $S$  is considered, then an upper bound on the number of subfunctions equals  $3k + 2n$ . For the concentration factor  $C$  the number of IBC subfunctions is always less or equal to  $3n + 2k$ . For large  $k$  and  $n$  the analytical approach to determining the IBC may be too expensive. Then a purely numerical SFS approximation, for example by the polygon inflation [27], may seem more appropriate.

### 5.3. Time and computer storage complexity of the IBC algorithm

Computational complexity theory is concerned with the required computer resources for the solution of a problem, mainly the computing time and required computer storage. We discuss these issues by using basic concepts of the complexity theory and algorithms, for example see [31, 8].

First, computer storage is not a problematic resource for the IBC algorithm because it scales only linearly with the dimension of the input data. Roughly, for each interval with constant contact change values ten parameters are to be stored in a table. Second, the time complexity class of the IBC calculation is determined by the choice of the algorithm for solving the contact problem, namely the convex hull algorithm or a comparable line-polygon intersection algorithm.

**Theorem 1.** *Let  $\text{alg}(n)$  be the time complexity class of a convex hull algorithm or a line-polygon intersection algorithm. Further, let  $m_I$  be the number of vertices of  $\mathcal{I}$  and  $m_{\mathcal{F}}$  be the number of vertices of  $\mathcal{F}$ . Then the time complexity class of the IBC algorithm is*

$$O(m_{\mathcal{F}} \text{alg}(m_I) + m_I (\text{alg}(m_I) + \text{alg}(m_{\mathcal{F}})) + (m_I + m_{\mathcal{F}}) \log(m_I + m_{\mathcal{F}})).$$

*If only  $m$  with  $m = \max(m_{\mathcal{F}}, m_I)$  is known, then the time complexity class of the IBC algorithm is*

$$O(m \text{alg}(m) + m \log(m)).$$

Therein, the big  $O$  notation, which is also called the Landau notation, serves to describe the asymptotic growth behavior. The proof of Thm. 1 is given in the Appendix D. The selection of the convex hull algorithm and of the line-polygon intersection algorithm significantly influence the asymptotic computing time. This was the reasoning for providing the pseudo-code of both versions in Appendix C. In our implementation the convex hull algorithm uses the Quickhull algorithm as implemented in MATLAB [18]. For the sake of simplicity we assume that the balance conditions hold and that all points are processed. Then the complexity class of the Quickhull algorithm equals  $O(m \log(m))$  for  $m$  points, see [4]. This results in the time complexity class  $O(m^2 \log(m))$ .  $m$  can be estimated to be less or equal than the maximal dimension of the matrix  $D$ .

#### 5.4. Intersection of the outer polygon with the IBC

As explained in Section 4.4, the SFS is the intersection of the outer polygon  $\mathcal{F}$  and the exterior of the IBC, which is the unbounded set if the IBC is considered to split the plane  $\mathbb{R}^2$  into two sets (namely a set around the origin which is enclosed by the IBC and its unbounded complement). The SFS contains the full information on the factor ambiguity of the nonnegative matrix factorization problem. If the IBC is determined, then the SFS can easily be computed numerically. However, the IBC also devises an analytical route for a precise (analytical) representation of the SFS. This approach is explained next.

Each edge of the outer polygon  $\mathcal{F}$  is given by an affine line of the form  $e_1x + e_2y = -1$  with proper parameters  $e_1$  and  $e_2$ . Furthermore, let us substitute  $z = \tan(\alpha)$  in Equation (11). Then the precise coordinates of a point of intersection of an edge of  $\mathcal{F}$  with the IBC are given by the solution of the equation

$$e_1p_1(z) + e_2p_2(z) = -1$$

if the considered subfunction of the inner boundary curve and the considered edge of  $\mathcal{F}$  intersect. The solution can be derived by solving the quadratic equation

$$0 = (e_1c_{x2} + e_2c_{y2} + c_{d2})z^2 + (e_1c_{x1} + e_2c_{y1} + c_{d1})z + (c_{x0} + c_{y0} + c_{d0}).$$

The result can be interpreted as Borgen-3, 3-triangles from [24]. We note that the advantages of this approach are limited to certain problematic cases. In general, the numerical approach is much faster and some minor loss of precision is not meaningful.

#### 5.5. A conic section approach to the IBC

The IBC representation (11) has a parametric form. It is also possible to represent it in the form of an implicit equation. This allows us to prove an interesting geometric property, namely that the IBC is composed of conic sections, see Figure 14.

**Theorem 2.** *Each of the IBC subfunctions is a segment of a conic section (namely an ellipse or circle, parabola or hyperbola or also a degenerate conic section as a line segment or a point).*

The proof is given in Appendix E. This theorem has some intriguing implications, as it places the IBC construction in a classical branch of mathematics. Possibly, this can provide more insight into the problem and to prove additional mathematical properties. If the conic section is non-degenerate (not just a line or single point), then the coefficients  $o_l$  and  $o_r$  lie on the conic section defined by the inner boundary curve subfunction. Further, the intersection of the lines that are given by the coefficients  $u_l$  and  $u_r$  (if it exists), also lies on the conic section. This can be proved by inserting them in Equation (14).

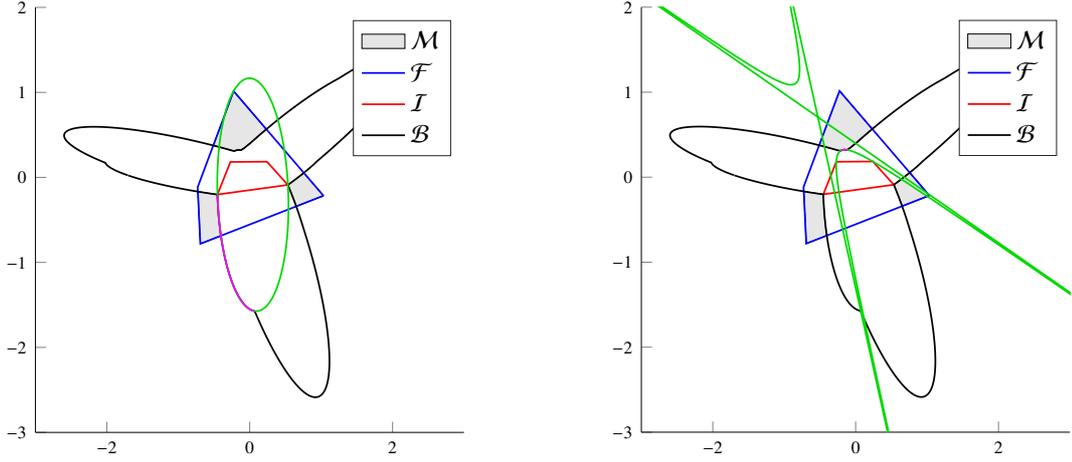


Figure 14: An ellipse (left, green) and a hyperbola (right, green) are drawn. They contribute in certain intervals to the inner boundary curve (magenta).

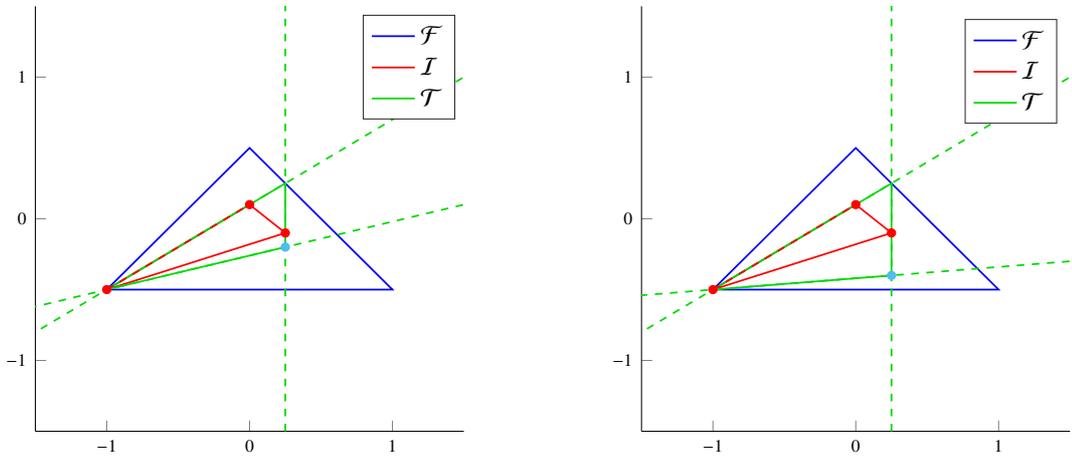


Figure 15: If a vertex of the inner polygon coincides with a vertex of the outer polygon, then an ambiguous free vertex can occur. Two possible free vertices (cyan dots in the left and right plot) are shown for possible Borgen-2, 3-triangles  $\mathcal{T}$  between the fixed polygons  $\mathcal{I}$  and  $\mathcal{F}$ .

### 5.6. Poorly conditioned cases of a numerical implementation of the IBC algorithm

The IBC algorithm is by its nature a precise analytical approach. However, the numerical and graphical representations require that the IBC is evaluated numerically. Some geometrical situations can result in poorly conditioned numerical computations. This is more or less inevitable for the suggested and other algorithms working with triangle constructions and rotations. Most of these cases occur if a vertex of  $\mathcal{I}$  is located very close to  $\mathcal{F}$  or if a vertex of  $\mathcal{I}$  coincides with a vertex of  $\mathcal{F}$ . To discuss this point, we recall that the triangle construction for finding the IBC parameters consists of finding tangents of the inner polygon through a given point of the outer polygon. If a vertex of  $\mathcal{I}$  is located close to  $\mathcal{F}$ , then this tangent can be defined by two points that are close to each other. Therefore, a small numerical error in one of those points can result in a large error for the tangent and for the IBC coordinates. In numerical mathematics such a problem is called ill-conditioned and so is the construction of certain triangles if vertices of  $\mathcal{I}$  are close to those of  $\mathcal{F}$ . In particular, if single-point or line-shaped subsets of SFS are expected, then numerical approximation methods, such as polygon inflation [27] should be preferred.

The limit case of coinciding vertices of  $\mathcal{I}$  and  $\mathcal{F}$  can occur if the spectral data matrix  $D$  contains multiple zero entries. Then Borgen-2, 3-triangles are not necessarily unique as illustrated in Figure 15. We assume that this case corresponds to a degenerated cone intersection with line shape. The suggested algorithmic implementation of the IBC construction cannot deal with this special situation and constructs only a single point. The algorithmic extension would be a linear interpolation in order to close gaps in the IBC curve which originate from this special geometric constellation. Further, we must expect a reduced numerical accuracy of a possible numerical implementation of the IBC algorithm in the case of coinciding vertices of  $\mathcal{I}$  and  $\mathcal{F}$ . This problem cannot be avoided easily.

## 6. Conclusion

The analytical boundary curve construction is not only an absolutely precise method in order to determine the set of feasible solutions for a nonnegative matrix factorization problem with three chemical species, but can also be practically implemented by the suggested algorithm. Closed-form expressions for the coordinates of the IBC have been derived which have revealed that the IBC piecewise consists of conic sections. This result joins the chemometric factor ambiguity problem with the classical field of mathematical geometry.

The IBC algorithm can support theoretical and analytical investigations with a focus on special and limit cases. For practical applications on high-dimensional experimental data the IBC algorithm is still applicable, but a numerical approximation approach might prove to be more robust. Nevertheless the IBC algorithm is promising in both the computing time and the precision of the results. The MATLAB Code of our implementation can be found on the FACPACK homepage [29]. This program code computes the list of sorted contact change values for a given (chemical data) matrix and provides a MATLAB function which enables the user to evaluate the boundary of the SFS with a high precision.

## Appendices

### A. Model matrix

We consider the following  $9 \times 8$  nonnegative model matrix with the rank 3. This matrix has many nonnegative factorizations with factors also of the rank 3.

$$D = \begin{pmatrix} 54 & 53 & 129 & 169 & 110 & 70 & 42 & 42 \\ 33 & 41 & 63 & 75 & 48 & 43 & 46 & 53 \\ 96 & 222 & 106 & 65 & 61 & 111 & 142 & 151 \\ 69 & 116 & 74 & 54 & 30 & 91 & 172 & 213 \\ 69 & 97 & 139 & 166 & 113 & 86 & 64 & 64 \\ 42 & 80 & 72 & 75 & 57 & 49 & 34 & 29 \\ 42 & 62 & 82 & 96 & 66 & 52 & 40 & 40 \\ 45 & 89 & 55 & 43 & 32 & 55 & 78 & 89 \\ 45 & 66 & 80 & 90 & 60 & 57 & 60 & 67 \end{pmatrix}. \quad (13)$$

### B. Coefficients of the IBC for Equation (11)

The following eight coefficients underlying the IBC coordinates as given in (11) have the following form:

$$\begin{aligned} c_{x0} &= -o_{11}o_{r1}u_{11}u_{r2}v_1 + o_{11}o_{r1}u_{12}u_{r1}v_1 + o_{11}o_{r2}u_{12}u_{r2}v_1 + o_{11}u_{11}u_{r2}v_1^2 - o_{12}o_{r1}u_{12}u_{r2}v_1 \\ &\quad + o_{12}u_{12}u_{r2}v_1^2 - o_{r1}u_{12}u_{r1}v_1^2 - o_{r2}u_{12}u_{r2}v_1^2 + o_{11}o_{r1}u_{12} - o_{11}o_{r1}u_{r2} + o_{11}u_{r2}v_1 - o_{r1}u_{12}v_1, \\ c_{x1} &= -o_{11}o_{r1}u_{11}u_{r2}v_2 + o_{11}o_{r1}u_{12}u_{r1}v_2 - o_{11}o_{r2}u_{11}u_{r2}v_1 + o_{11}o_{r2}u_{12}u_{r2}v_2 + 2o_{11}u_{11}u_{r2}v_1v_2 \\ &\quad + o_{12}o_{r1}u_{12}u_{r1}v_1 - o_{12}o_{r1}u_{12}u_{r2}v_2 + 2o_{12}u_{12}u_{r2}v_1v_2 - 2o_{r1}u_{12}u_{r1}v_1v_2 - 2o_{r2}u_{12}u_{r2}v_1v_2 \\ &\quad - o_{11}o_{r1}u_{11} + o_{11}o_{r1}u_{r1} + o_{11}o_{r2}u_{12} + o_{11}u_{11}v_1 + o_{11}u_{r2}v_2 - o_{12}o_{r1}u_{r2} + o_{12}u_{12}v_1 \\ &\quad + o_{12}u_{r2}v_1 - o_{r1}u_{12}v_2 - o_{r1}u_{r1}v_1 - o_{r2}u_{12}v_1 - o_{r2}u_{r2}v_1 + o_{11} - o_{r1}, \\ c_{x2} &= -o_{11}o_{r2}u_{11}u_{r2}v_2 + o_{11}u_{11}u_{r2}v_2^2 + o_{12}o_{r1}u_{12}u_{r1}v_2 + o_{12}u_{12}u_{r2}v_2^2 - o_{r1}u_{12}u_{r1}v_2^2 \\ &\quad - o_{r2}u_{12}u_{r2}v_2^2 - o_{11}o_{r2}u_{11} + o_{11}u_{11}v_2 + o_{12}o_{r1}u_{r1} + o_{12}u_{12}v_2 + o_{12}u_{r2}v_2 - o_{r1}u_{r1}v_2 \\ &\quad - o_{r2}u_{12}v_2 - o_{r2}u_{r2}v_2 + o_{12} - o_{r2}, \\ c_{y0} &= -o_{11}o_{r2}u_{11}u_{r2}v_1 - o_{11}u_{11}u_{r1}v_1^2 + o_{12}o_{r1}u_{12}u_{r1}v_1 - o_{12}u_{12}u_{r1}v_1^2 + o_{r1}u_{11}u_{r1}v_1^2 \\ &\quad + o_{r2}u_{11}u_{r2}v_1^2 - o_{11}o_{r2}u_{r2} - o_{11}u_{11}v_1 - o_{11}u_{r1}v_1 + o_{12}o_{r1}u_{12} - o_{12}u_{12}v_1 + o_{r1}u_{11}v_1 \\ &\quad + o_{r1}u_{r1}v_1 + o_{r2}u_{r2}v_1 - o_{11} + o_{r1}, \\ c_{y1} &= o_{11}o_{r2}u_{11}u_{r1}v_1 - o_{11}o_{r2}u_{11}u_{r2}v_2 - 2o_{11}u_{11}u_{r1}v_1v_2 - o_{12}o_{r1}u_{11}u_{r1}v_1 + o_{12}o_{r1}u_{12}u_{r1}v_2 \\ &\quad - o_{12}o_{r2}u_{11}u_{r2}v_1 + o_{12}o_{r2}u_{12}u_{r1}v_1 - 2o_{12}u_{12}u_{r1}v_1v_2 + 2o_{r1}u_{11}u_{r1}v_1v_2 + 2o_{r2}u_{11}u_{r2}v_1v_2 \\ &\quad + o_{11}o_{r2}u_{r1} - o_{11}u_{11}v_2 - o_{11}u_{r1}v_2 - o_{12}o_{r1}u_{11} + o_{12}o_{r2}u_{12} - o_{12}o_{r2}u_{r2} - o_{12}u_{12}v_2 \\ &\quad - o_{12}u_{r1}v_1 + o_{r1}u_{11}v_2 + o_{r1}u_{r1}v_2 + o_{r2}u_{11}v_1 + o_{r2}u_{r2}v_2 - o_{12} + o_{r2}, \\ c_{y2} &= o_{11}o_{r2}u_{11}u_{r1}v_2 - o_{11}u_{11}u_{r1}v_2^2 - o_{12}o_{r1}u_{11}u_{r1}v_2 - o_{12}o_{r2}u_{11}u_{r2}v_2 + o_{12}o_{r2}u_{12}u_{r1}v_2 \\ &\quad - o_{12}u_{12}u_{r1}v_2^2 + o_{r1}u_{11}u_{r1}v_2^2 + o_{r2}u_{11}u_{r2}v_2^2 - o_{12}o_{r2}u_{11} + o_{12}o_{r2}u_{r1} - o_{12}u_{r1}v_2 + o_{r2}u_{11}v_2, \\ c_{d0} &= o_{11}o_{r2}u_{12}u_{r2} + o_{11}u_{12}u_{r1}v_1 - o_{12}o_{r1}u_{12}u_{r2} + o_{12}u_{12}u_{r2}v_1 - o_{r1}u_{11}u_{r2}v_1 - o_{r2}u_{12}u_{r2}v_1 \\ &\quad + u_{11}u_{r2}v_1^2 - u_{12}u_{r1}v_1^2 + o_{11}u_{12} - o_{r1}u_{r2} - u_{12}v_1 + u_{r2}v_1, \end{aligned}$$

$$\begin{aligned}
c_{d1} = & -o_{l1}o_{r2}u_{l1}u_{r2} - o_{l1}o_{r2}u_{l2}u_{r1} - o_{l1}u_{l1}u_{r1}v_1 + o_{l1}u_{l2}u_{r1}v_2 + o_{l2}o_{r1}u_{l1}u_{r2} + o_{l2}o_{r1}u_{l2}u_{r1} \\
& - o_{l2}u_{l1}u_{r2}v_1 + o_{l2}u_{l2}u_{r2}v_2 + o_{r1}u_{l1}u_{r1}v_1 - o_{r1}u_{l1}u_{r2}v_2 + o_{r2}u_{l2}u_{r1}v_1 - o_{r2}u_{l2}u_{r2}v_2 \\
& + 2u_{l1}u_{r2}v_1v_2 - 2u_{l2}u_{r1}v_1v_2 - o_{l1}u_{l1} \\
& + o_{l2}u_{l2} + o_{r1}u_{r1} - o_{r2}u_{r2} + u_{l1}v_1 - u_{l2}v_2 - u_{r1}v_1 + u_{r2}v_2, \\
c_{d2} = & o_{l1}o_{r2}u_{l1}u_{r1} - o_{l1}u_{l1}u_{r1}v_2 - o_{l2}o_{r1}u_{l1}u_{r1} - o_{l2}u_{l1}u_{r2}v_2 + o_{r1}u_{l1}u_{r1}v_2 + o_{r2}u_{l2}u_{r1}v_2 \\
& + u_{l1}u_{r2}v_2^2 - u_{l2}u_{r1}v_2^2 - o_{l2}u_{l1} + o_{r2}u_{r1} + u_{l1}v_2 - u_{r1}v_2.
\end{aligned}$$

### C. Pseudocode for determining contact changes values

The following five pseudocode elements serve to determine the contact change values of the first up to the third kind. For explanations and details see Section 4. An implementation in MATLAB using the convex hull approach can be found on the FACPAC homepage [29].

---

#### Algorithm 1 Computing $\alpha^1$

---

**Require:**  $\mathcal{F}_D$ .

**Ensure:** A list of contact change values of the first kind  $\alpha_i^1$  with the corresponding parameters  $(v)_i$ ,  $i = 1, \dots, m_{\mathcal{I}}$ .

- 1: Set  $i = 1$ .
  - 2: **for**  $t$  as a vertex of  $\mathcal{F}_D$  **do**
  - 3:   Set  $\alpha_i^1 = \text{atan2}(t)$ .
  - 4:   Save the edge of  $\mathcal{F}_D$  that follows  $i$  counter-clockwise as  $(v)_i$ .
  - 5:   Set  $i = i + 1$ .
  - 6: **end for**
- 

---

#### Algorithm 2 Computing $\alpha^2$ with the convex hull

---

**Require:**  $\mathcal{F}, \mathcal{I}$ .

**Ensure:** A list of the left contact change values of the second kind  $\alpha_i^{2l}$  with the corresponding parameters  $(u)_i$  and a list of the right contact change values of the second kind  $\alpha_i^{2r}$  with the corresponding parameters  $(u_r)_i$ ,  $i = 1, \dots, m_{\mathcal{F}}$ .

- 1: Set  $i = 1$ .
  - 2: **for**  $s_{lr}$  as a vertex of  $\mathcal{F}$  **do**
  - 3:   Find the left tangent  $t_l$  and the right tangent  $t_r$  of  $\mathcal{I}$  from  $s_{lr}$  with a convex hull algorithm. (The orientation is towards the origin.)
  - 4:   Set  $\alpha_i^{2l} = \text{atan2}(t_l)$ .
  - 5:   Save the edge of  $\mathcal{F}$  that follows  $s_{lr}$  counter-clockwise as  $(u)_i$ .
  - 6:   Set  $\alpha_i^{2r} = \text{atan2}(t_r)$ .
  - 7:   Save the edge of  $\mathcal{F}$  that follows  $s_{lr}$  counter-clockwise as  $(u_r)_i$ .
  - 8:   Set  $i = i + 1$ .
  - 9: **end for**
- 

---

#### Algorithm 3 Computing $\alpha^2$ by line-polygon intersection

---

**Require:**  $\mathcal{F}_d, \mathcal{I}_d$ .

**Ensure:** A list of the left contact change values of the second kind  $\alpha_i^{2l}$  with the corresponding parameters  $(u)_i$  and list of the right contact change values of the second kind  $\alpha_i^{2r}$  with the corresponding parameters  $(u_r)_i$ ,  $i = 1, \dots, m_{\mathcal{F}}$ .

- 1: Set  $i = 1$ .
  - 2: **for**  $s_{lr}$  as a line that contains an edge of  $\mathcal{I}_d$  **do**
  - 3:   Find the left intersection  $t_l$  and the right intersection  $t_r$  of  $s_{lr}$  with  $\mathcal{F}_d$  using a line-polygon intersection algorithm.
  - 4:   Set  $\alpha_i^{2l} = \text{atan2}(t_l)$ .
  - 5:   Save the right vertex of the chosen edge of  $\mathcal{I}_d$  as  $(u)_i$ .
  - 6:   Set  $\alpha_i^{2r} = \text{atan2}(t_r)$ .
  - 7:   Save the right vertex of the chosen edge of  $\mathcal{I}_d$  as  $(u_r)_i$ .
  - 8:   Set  $i = i + 1$ .
  - 9: **end for**
-

---

**Algorithm 4** Computing  $\alpha^3$  by a convex hull approach

---

**Require:**  $\mathcal{F}, \mathcal{I}, \mathcal{F}_d, \mathcal{I}_d$ .**Ensure:** A list of the left contact change values of the third kind  $\alpha_i^{3l}$  with the corresponding parameters  $(o_l)_i$  and a list of the right contact change values of the third kind  $\alpha_i^{3r}$  with the corresponding parameters  $(o_r)_i$ ,  $i = 1, \dots, m_I$ .

- 1: Set  $i = 1$ .
  - 2: **for**  $q_{lr}$  as a vertex of  $\mathcal{F}_d$  **do**
  - 3: Find the left tangent  $s_r$  and the right tangent  $s_l$  from  $q_{lr}$  to  $\mathcal{I}_d$  with a convex hull algorithm.
  - 4: Find the left tangent  $q_{lr}$  and the right tangent  $t_l$  from  $s_l$  to  $\mathcal{I}$  with a convex hull algorithm.
  - 5: Set  $\alpha_i^{3l} = \text{atan2}(t_l)$ .
  - 6: Save the edge of  $\mathcal{F}_d$  that follows  $q_{lr}$  counter-clockwise as  $(o_l)_i$ .
  - 7: Find the left tangent  $t_r$  and the right tangent  $q_{lr}$  from  $s_r$  to  $\mathcal{I}$  with a convex hull algorithm.
  - 8: Set  $\alpha_i^{3r} = \text{atan2}(t_r)$ .
  - 9: Save the edge of  $\mathcal{F}_d$  that follows  $q_{lr}$  counter-clockwise as  $(o_r)_i$ .
  - 10: Set  $i = i + 1$ .
  - 11: **end for**
- 

---

**Algorithm 5** Computing  $\alpha^3$  by a line-polygon approach

---

**Require:**  $\mathcal{F}, \mathcal{I}, \mathcal{F}_d, \mathcal{I}_d$ .**Ensure:** A list of the left contact change values of the third kind  $\alpha_i^{3l}$  with the corresponding parameters  $(o_l)_i$  and a list of the right contact change values of the third kind  $\alpha_i^{3r}$  with the corresponding parameters  $(o_r)_i$ ,  $i = 1, \dots, k_I$ .

- 1: Set  $i = 1$
  - 2: **for**  $q_{lr}$  as a line that contains an edge of  $\mathcal{I}$  **do**
  - 3: Find the left intersection  $s_r$  and the right intersection  $s_l$  of  $q_{lr}$  with  $\mathcal{F}$  using a line-polygon intersection algorithm
  - 4: Find the right intersection  $t_l$  of  $s_l$  with  $\mathcal{F}_d$  using a line-polygon intersection algorithm (the left intersection is  $q_{lr}$ )
  - 5: Set  $\alpha_i^{3l} = \text{atan2}(t_l)$
  - 6: Save the right vertex of the chosen edge of  $\mathcal{I}$  as  $(o_l)_i$
  - 7: Find the left intersection  $t_r$  of  $s_2$  with  $\mathcal{F}_d$  using a line-polygon intersection algorithm (the right intersection is  $q_{lr}$ )
  - 8: Set  $\alpha_i^{3r} = \text{atan2}(t_r)$
  - 9: Save the right vertex of the chosen edge of  $\mathcal{I}$  as  $(o_r)_i$
  - 10: Set  $i = i + 1$
  - 11: **end for**
- 

**D. Proof of Theorem 1**

*Proof.* Each step of the construction of the IBC is considered concerning its computing time complexity, starting with the computation of the contact change values.

To compute the contact change values of the first kind,  $m_I$  iterations with a constant number of operations are performed. Therefore, this step has the time complexity class  $\mathcal{O}(m_I)$ .

To determine the contact change values of the second kind,  $m_{\mathcal{F}}$  iterations are performed. Each of the iterations consists of either a convex hull algorithm over  $m_I + 1$  vectors or a line-polygon intersection algorithm over  $m_I + 1$  lines and a constant number of further operations. Thus, this step has the time complexity class  $\mathcal{O}(m_{\mathcal{F}} \text{alg}(m_I))$ .

To compute the contact change values of the third kind,  $m_I$  iterations are performed. Each of the iterations consists of either a convex hull algorithm over  $m_{\mathcal{F}} + 1$  vectors and two more repetitions of a convex hull algorithm over  $m_I + 1$  vectors or a line-polygon intersection algorithm over  $m_{\mathcal{F}} + 1$  lines and two more repetitions of a line-polygon intersection algorithm over  $m_I + 1$  lines. There is also a constant number of further operations. Therefore,  $\mathcal{O}(m_I(\text{alg}(m_I) + \text{alg}(m_{\mathcal{F}})))$  is the time complexity class of this step.

Further, the sorting of the  $3m_I + 2m_{\mathcal{F}}$  contact change values is required. A good sorting algorithm has the time complexity of  $\mathcal{O}(n \log(n))$ , see [8]. Thus, this step has the time complexity class  $\mathcal{O}((m_I + m_{\mathcal{F}}) \log(m_I + m_{\mathcal{F}}))$ .

Finally, the resulting functions have to be calculated for each of the  $3m_I + 2m_{\mathcal{F}}$  subintervals. This requires a constant number of operations for each iteration. Therefore, this step has the time complexity class  $\mathcal{O}(3m_I + 2m_{\mathcal{F}})$ .

The resulting time complexity of the IBC algorithm is given by

$$\mathcal{O}(m_{\mathcal{F}} \text{alg}(m_I) + m_I(\text{alg}(m_I) + \text{alg}(m_{\mathcal{F}})) + (m_I + m_{\mathcal{F}}) \log(m_I + m_{\mathcal{F}})).$$

The second result follows from this. □

## E. Proof of Theorem 2

*Proof.* The theorem is proven by bringing the parametric curve by Equation (11) in the general Cartesian form

$$d_{x2}x^2 + d_{xy}xy + d_{y2}y^2 + d_x x + d_y y + d_0 = 0. \quad (14)$$

This equation defines a conic section problem and all possible cases of conic sections can be described in this way. Due to the complexity of the expressions we prefer to work with the symbolic algebra program tool Maple and its `eliminate` and `collect` functions. The coefficients of the Cartesian form (14) are given by

$$\begin{aligned} d_{x2} &= c_{y0}^2 c_{d2}^2 - c_{y0} c_{y1} c_{d1} c_{d2} - 2c_{y0} c_{y2} c_{d0} c_{d2} + c_{y0} c_{y2} c_{d1}^2 + c_{y1}^2 c_{d0} c_{d2} - c_{y1} c_{y2} c_{d0} c_{d1} + c_{y2}^2 c_{d0}^2, \\ d_{y2} &= c_{x0}^2 c_{d2}^2 - c_{x0} c_{x1} c_{d1} c_{d2} - 2c_{x0} c_{x2} c_{d0} c_{d2} + c_{x0} c_{x2} c_{d1}^2 + c_{x1}^2 c_{d0} c_{d2} - c_{x1} c_{x2} c_{d0} c_{d1} + c_{x2}^2 c_{d0}^2, \\ d_x &= c_{x0} c_{y1} c_{y2} c_{d1} + c_{x1} c_{y0} c_{y1} c_{d2} + c_{x1} c_{y1} c_{y2} c_{d0} + c_{x2} c_{y0} c_{y1} c_{d1} + 2c_{x0} c_{y0} c_{y2} c_{d2} - c_{x0} c_{y1}^2 c_{d2} \\ &\quad - 2c_{x0} c_{y2}^2 c_{d0} - 2c_{x1} c_{y0} c_{y2} c_{d1} - 2c_{x2} c_{y0}^2 c_{d2} + 2c_{x2} c_{y0} c_{y2} c_{d0} - c_{x2} c_{y1}^2 c_{d0}, \\ d_y &= -2c_{x0}^2 c_{y2} c_{d2} + c_{x0} c_{x1} c_{y1} c_{d2} + c_{x0} c_{x1} c_{y2} c_{d1} + 2c_{x0} c_{x2} c_{y0} c_{d2} - 2c_{x0} c_{x2} c_{y1} c_{d1} \\ &\quad + 2c_{x0} c_{x2} c_{y2} c_{d0} - c_{x1}^2 c_{y0} c_{d2} - c_{x1}^2 c_{y2} c_{d0} + c_{x1} c_{x2} c_{y0} c_{d1} + c_{x1} c_{x2} c_{y1} c_{d0} - 2c_{x2}^2 c_{y0} c_{d0}, \\ d_{xy} &= -2c_{x0} c_{y0} c_{d2}^2 + c_{x0} c_{y1} c_{d1} c_{d2} + 2c_{x0} c_{y2} c_{d0} c_{d2} - c_{x0} c_{y2} c_{d1}^2 + c_{x1} c_{y0} c_{d1} c_{d2} - 2c_{x1} c_{y1} c_{d0} c_{d2} \\ &\quad + c_{x1} c_{y2} c_{d0} c_{d1} + 2c_{x2} c_{y0} c_{d0} c_{d2} - c_{x2} c_{y0} c_{d1}^2 + c_{x2} c_{y1} c_{d0} c_{d1} - 2c_{x2} c_{y2} c_{d0}^2, \\ d_0 &= c_{x0}^2 c_{y2}^2 + c_{x2}^2 c_{y0}^2 - c_{x0} c_{x1} c_{y1} c_{y2} - 2c_{x0} c_{x2} c_{y0} c_{y2} + c_{x0} c_{x2} c_{y1}^2 + c_{x1}^2 c_{y0} c_{y2} - c_{x1} c_{x2} c_{y0} c_{y1} \end{aligned}$$

where all coefficients are listed in Appendix B. These relations can be verified by observing that the parametric form  $(x, y) = (x(z), y(z))$  fulfills Equation (14) with  $z = \tan(\alpha)$ . □

## Declarations

Conflict of interest: The authors declare that they have no conflict of interest or other ethical conflicts concerning this paper.

## References

- [1] H. Abdollahi and R. Tauler. Uniqueness and rotation ambiguities in multivariate curve resolution methods. *Chemom. Intell. Lab. Syst.*, 108(2):100–111, 2011.
- [2] A. Aggarwal, H. Booth, J. O’Rourke, S. Suri, and C. K. Yap. Finding minimal convex nested polygons. *Inf. Comput.*, 83(1):98–110, 1989.
- [3] T. Andersons. Analytical boundary curve construction for the solution set of nonnegative matrix factorisations. Master’s thesis, Universität Rostock, 2020.
- [4] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. The Quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, 1996.
- [5] S. Beyramysoltan, H. Abdollahi, and R. Rajkó. Newer developments on self-modeling curve resolution implementing equality and unimodality constraints. *Anal. Chim. Acta*, 827(0):1–14, 2014.
- [6] O.S. Borgen and B.R. Kowalski. An extension of the multivariate component-resolution method to three components. *Anal. Chim. Acta*, 174:1–26, 1985.
- [7] R. Bro, E. Acar, and T. G. Kolda. Resolving the sign ambiguity in the singular value decomposition. *J. Chemom.*, 22(2):135–140, 2008.
- [8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, Cambridge, 2005.
- [9] A. Golshan, H. Abdollahi, S. Beyramysoltan, M. Maeder, K. Neymeyr, R. Rajkó, M. Sawall, and R. Tauler. A review of recent methods for the determination of ranges of feasible solutions resulting from soft modelling analyses of multivariate data. *Anal. Chim. Acta*, 911:1–13, 2016.
- [10] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 2012.
- [11] R.C. Henry. Duality in multivariate receptor models. *Chemom. Intell. Lab. Syst.*, 77(1-2):59–63, 2005.
- [12] A. Jürß, M. Sawall, and K. Neymeyr. On generalized Borgen plots. I: From convex to affine combinations and applications to spectral data. *J. Chemom.*, 29(7):420–433, 2015.
- [13] A. Jürß, M. Sawall, and K. Neymeyr. On generalized Borgen plots. II: The line-moving algorithm and its numerical implementation. *J. Chemom.*, 30:636–650, 2016.
- [14] C. Kubis, M. Sawall, A. Block, K. Neymeyr, R. Ludwig, A. Börner, and D. Selent. An operando FTIR spectroscopic and kinetic study of carbon monoxide pressure influence on rhodium-catalyzed olefin hydroformylation. *Chem.-Eur. J.*, 20(37):11921–11931, 2014.
- [15] W.H. Lawton and E.A. Sylvester. Self modelling curve resolution. *Technometrics*, 13:617–633, 1971.
- [16] M. Maeder and Y.M. Neuhold. *Practical data analysis in chemistry*. Elsevier, Amsterdam, 2007.
- [17] E.R. Malinowski. Window factor analysis: Theoretical derivation and application to flow injection analysis data. *J. Chemom.*, 6(1):29–40, 1992.
- [18] The Mathworks, Inc., Natick, Massachusetts. *MATLAB version 9.10.0.1613233 (R2021a)*, 2021.
- [19] A. Meister. Estimation of component spectra by the principal components method. *Anal. Chim. Acta*, 161:149–161, 1984.

- [20] K. Neymeyr and M. Sawall. On the set of solutions of the nonnegative matrix factorization problem. *SIAM J. Matrix Anal. Appl.*, 39:1049–1069, 2018.
- [21] K. Neymeyr, M. Sawall, and D. Hess. Pure component spectral recovery and constrained matrix factorizations: Concepts and applications. *J. Chemom.*, 24:67–74, 2010.
- [22] R. Rajkó. Natural duality in minimal constrained self modeling curve resolution. *J. Chemom.*, 20(3-4):164–169, 2006.
- [23] R. Rajkó. Studies on the adaptability of different Borgen norms applied in self-modeling curve resolution (SMCR) method. *J. Chemom.*, 23(6):265–274, 2009.
- [24] R. Rajkó and K. István. Analytical solution for determining feasible regions of self-modeling curve resolution (SMCR) method based on computational geometry. *J. Chemom.*, 19(8):448–463, 2005.
- [25] M. Sawall, A. Jürß, H. Schröder, and K. Neymeyr. *On the analysis and computation of the area of feasible solutions for two-, three- and four-component systems*, volume 30 of Data Handling in Science and Technology, “Resolving Spectral Mixtures”, Ed. C. Ruckebusch, chapter 5, pages 135–184. Elsevier, Cambridge, 2016.
- [26] M. Sawall, A. Jürß, H. Schröder, and K. Neymeyr. Simultaneous construction of dual Borgen plots. I: The case of noise-free data. *J. Chemom.*, 31:e2954, 2017.
- [27] M. Sawall, C. Kubis, D. Selent, A. Börner, and K. Neymeyr. A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. I: Concepts and applications. *J. Chemom.*, 27:106–116, 2013.
- [28] M. Sawall, A. Moog, C. Kubis, H. Schröder, D. Selent, R. Franke, A. Brächer, A. Börner, and K. Neymeyr. Simultaneous construction of dual Borgen plots. II: Algorithmic enhancement for applications to noisy spectral data. *J. Chemom.*, 32:e3012, 2018.
- [29] M. Sawall, A. Moog, and K. Neymeyr. FACPACK: A software for the computation of multi-component factorizations and the area of feasible solutions, Revision 1.3. FACPACK homepage: <http://www.math.uni-rostock.de/facpack/>, 2018.
- [30] M. Sawall and K. Neymeyr. A fast polygon inflation algorithm to compute the area of feasible solutions for three-component systems. II: Theoretical foundation, inverse polygon inflation, and FAC-PACK implementation. *J. Chemom.*, 28:633–644, 2014.
- [31] M. Sipser. *Introduction to the theory of computation*. Cengage Learning, Boston, 2012.