Evidence for mixed rationalities in preference formation
Supplementary Information: Fitting algorithm

Alexandru-Ionuţ Băbeanu and Diego Garlaschelli
Lorentz Institute for Theoretical Physics, Leiden University, The Netherlands
(Dated: October 19, 2017)

This document explains the procedure used for simultaneously tuning the α and β parameters of either of the two stochastic models of culture, such that a match is obtained between the model and the empirical data, in terms of the averages of the AIVD and SIVD observables:

$$\langle \mathrm{AIVD}(\alpha,\beta)\rangle = \mathrm{AIVD}_{\mathrm{emp}}, \qquad (1)$$
$$\langle \mathrm{SIVD}(\alpha,\beta)\rangle = \mathrm{SIVD}_{\mathrm{emp}},$$

for a fixed number of prototypes k, assuming that either of the two equalities above is satisfied when there is an overlap between the uncertainty range associated to the quantity on the left side and that associated to the quantity on the right side.

There are multiple reasons why this problem is challenging:

- an analytical formula for the $\langle \mathrm{SIVD}(\alpha,\beta)\rangle$ quantity could not be found
- although an analytical formula for the $\langle \mathrm{AIVD}(\alpha,\beta)\rangle$ quantity was found[1] (see main text), this formula does not allow for inverting the function and for analytically solving the system
- the $\langle \mathrm{SIVD}(\alpha,\beta)\rangle$, $\mathrm{AIVD}_{\mathrm{emp}}$ and $\mathrm{SIVD}_{\mathrm{emp}}$ quantities have non-vanishing uncertainty ranges attached to them

Assuming that there exists a unique solution to the above system, a numerical approach for solving it is in order. The method used here relies on a nested, 2-level, adapted bisection method. The first (inner) level of the method takes care of fitting, via bisection, the first quantity for a fixed β – it finds the α value for which $\langle \mathrm{AIVD}(\alpha,\beta)\rangle = \mathrm{AIVD}_{\mathrm{emp}}$ is satisfied for a given β. The second (outer) level of the method takes care of fitting, via bisection, the second quantity – it finds the β for which $\langle \mathrm{SIVD}(\alpha(\beta),\beta)\rangle = \mathrm{SIVD}_{\mathrm{emp}}$ is satisfied, where $\alpha(\beta)$ is provided by the first level. This choice of assigning the AIVD and SIVD observables and the α and β parameters to the two levels in this manner is numerically convenient for several reasons. First, the AIVD can be much more easily computed via the analytical formula, such that assigning it to the first level, which is repeated multiple times (once for each value of β that the second level samples) is more effective. Second, the model AIVD turns out to be relatively insensitive to β for relatively many combinations of values for the k and α parameters, such that fitting AIVD in terms of α within the first level makes more sense.

In addition to adaptations required by the 2-level scheme, other adaptations with respect to the traditional bisection method are needed for allowing it to work with model and empirical uncertainties, as well as to enhance the numerical precision for the $\langle \mathrm{SIVD}(\alpha,\beta)\rangle$ quantity when needed, to the extent needed. Moreover, in addition to statistical errors originating directly in the empirical uncertainties of the $\mathrm{AIVD}_{\mathrm{emp}}$ and $\mathrm{SIVD}_{\mathrm{emp}}$ quantities and in the numerical uncertainty of the model SIVD quantity, the second level of the method is also affected by "systematic errors" on $\langle \mathrm{SIVD}(\alpha(\beta),\beta)\rangle$, originating in the fitting procedure at the first level, and indirectly in the empirical uncertainty of $\mathrm{AIVD}_{\mathrm{emp}}$ – which for all practical purposes can be assumed fixed, thus motivating using the term "systematic" for its propagation to the model SIVD at the second level.

In order to address all these challenges in a self consistent way, the method developed here turns out to be quite sophisticated, which is why it is explained in detail in the following four sections. Specifically, Sec. A focuses on the first fitting level, Sec. B focuses on the second fitting level, Sec. C describes how various subproblems invoked by the previous two sections are addressed, while Sec. D describes how the tools presented in Sections A, B and C are used for producing some of the results in the main text. The method is potentially of use for addressing other problems that are formally similar to the problem presented here, although certain adaptations might be needed.

Since the method has mostly an algorithmic nature, much of it is explained via pseudocode, such that a few conventions that will be extensively used below and that are not necessarily standard are worth mentioning. First, the "=" symbol is used with double meaning: in a normal statement (such as "$a = b$") it is to be interpreted as an assignment (of the value of variable b to variable a); in the header of an **if** or **while** statement (such as "**if** $a = b$") it is to be interpreted as a check (of whether the values of a and b are equal). A variable is implicitly declared when it first appears, either on the left side of an assignment or in the header of a function definition (in which case it is also called an argument or function parameter); the scope of the variable is the part of the function below and to the right of the place where it first appears. Functions are distinguished from each other through their names, their numbers of arguments and

[1] Which implies that the specific uncertainty range of $\langle \mathrm{AIVD}(\alpha,\beta)\rangle$ has a null width.

the types of those arguments [2] On the other hand, the arguments of a function are distinguished from each other via their order. Some variables are actually ordered sequences of other variables, which in turn are denoted by $(x_1, .., x_n)$ notation. In the same spirit, an assignments of the type $X = (x_1, .., x_n)$ is referred to as a "variable compression", while one of the type $(x_1, .., x_n) = X$ is referred to as a "variable decompression". These allow for keeping the pseudocode compact, while still rigorous. An uncertainty range refers to an interval $[x-\delta x, x+\delta x]$, where x is a mean and δx is an error relying (directly, or indirectly) on a standard mean error calculation, the uncertainty range being formally encoded by the sorted $(x, \delta x)$ sequence. Note that the square brackets "[,]" are consistently used to denote an interval of real numbers, while the round brackets "(,)" are used to denote an ordered sequence of two or more elements. Finally, it is worth noting that the pseudocode relies heavily on function calls and on recursive definitions, and that there is a certain parallelism between the functions defined in Sec. A and those defined in Sec. B.

A. First level fitting

This section presents the algorithm part concerned with the first fitting level. The algorithm is split in three main functions: Fit-1, Bisect-1, Displace-1, all of them returning the same type of information. Fit-1 always calls Bisect-1, while the latter may or may not call Displace-1 at any stage, which in turn may or may not call Bisect-1. The pseudocode also invokes two constants, which are assumed to be known a-priori and available for use anywhere in these three functions. The first constant is $\delta\alpha$, which controls the desired resolution ($\delta\alpha$ is essentially a grid-spacing) in the α parameter, which is here set to the inverse of the number of features: $\delta\alpha = \frac{1}{F}$ [3]. The second constant is AIVD_{emp}, which stands for the AIVD uncertainty range for the empirical data.

Function Fit-1 acts as an interface for the first-level fitting, which consists of tuning the α parameter, for given values of β and k, such that the AIVD quantity matches the empirical value. Here, β is a real number belonging to $[0, 1]$ while k is a strictly positive integer number. The method returns the left (α_L) and right (α_R) margins of the tightest α interval found, together with the estimated α match within this interval assuming linearity (α_{fit}) and an associated error (α_{err}). It assumes that the empirical AIVD can actually be uniquely matched by varying α, for the given values of β and k. The method essentially carries out some initializations (Lines 2,3), before passing the task to Bisect-1.

```
1: function Fit-1(β, k)
2:     (α_L, α_R) = Init-1(δα)                              ▷ initializing the α-interval
3:     AIVD_L = ⟨AIVD⟩^k_{α_L,β}; AIVD_R = ⟨AIVD⟩^k_{α_R,β}   ▷ analytic calculations (see main text)
4:     return Bisect-1(α_L, α_R, AIVD_L, AIVD_R, β, k)
5: end function
```

Function Bisect-1 is mostly a typical, recursive implementation of the bisection method. This sequentially narrows down the $[\alpha_L, \alpha_R]$ interval, such that at each stage the empirical AIVD is contained, namely that $\min(\text{AIVD}_L, \text{AIVD}_R) < \text{AIVD}_{\text{emp}} < \max(\text{AIVD}_L, \text{AIVD}_R)$ is satisfied, where the $\text{AIVD}_L, \text{AIVD}_R$ values correspond to the left and right margins of the α interval. Here, $\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R$ are real numbers belonging to $[0, 1]$ while β and k are of the same type as in Fit-1. It returns the same type of information as Fit-1. The method converges, the fitting being considered complete, when the interval has reached the $\delta\alpha$ resolution limit, in which case estimations for an "ideal" α inside this interval α_{fit} and its error α_{err} are made and returned together with the boundaries of the interval (lines 3-6). Moreover, the method may also call Displace-1 in case the AIVD_M value corresponding to the computed midpoint α_M happens to fall within the AIVD_{emp} uncertainty range (lines 8-10) – this is needed in order to keep the output format consistent and the final α interval relatively narrow. Otherwise, the method decides to zoom in (by calling itself) on either the left or right halves of the interval, depending on the position of AIVD_{emp} with respect to AIVD_L, AIVD_M and AIVD_R (lines 11-16).

[2] Sometimes this can be confusing, since the types of the arguments are only mentioned in the text before the definition of the function. In these cases however, the reader is guided by the names of the arguments, which in the function definition are kept as close as possible to those in the function call(s).

[3] There is no clear lower bound on $\delta\alpha$, regardless of which stochastic model is used, but $\frac{1}{F}$ is a lower bound on $\delta\beta$ when PG is used, so for simplicity the choice $\delta\alpha = \delta\beta = \frac{1}{F}$ is made.

function BISECT-1($\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \beta, k$)
  $\alpha_M = \text{MIDDLE}(\alpha_L, \alpha_R, \delta\alpha)$ ▷ computing midpoint on the α grid
  if $\neg$DISTINCT($\alpha_M, \alpha_L, \alpha_R$) **then**
    $(\alpha_{\text{fit}}, \alpha_{\text{err}}) = \text{INTERNFITLIN-1}(\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \text{AIVD}_{\text{emp}})$
    return $(\alpha_L, \alpha_R, \alpha_{\text{fit}}, \alpha_{\text{err}})$ ▷ fitting complete
  end if
  $\text{AIVD}_M = \langle\text{AIVD}\rangle^k_{\alpha_M,\beta}$ ▷ analytic calculations (see main text)
  if MATCH-1($\text{AIVD}_M, \text{AIVD}_{\text{emp}}$) **then**
    return DISPLACE-1($\alpha_L, \alpha_R, \alpha_M, \text{AIVD}_L, \text{AIVD}_R, \beta, k$)
  end if
  if ORD-1($\text{AIVD}_L, \text{AIVD}_R$) = ORD-1($\text{AIVD}_M, \text{AIVD}_{\text{emp}}$) **then**
    $\alpha_L = \alpha_M$; $\text{AIVD}_L = \text{AIVD}_M$ ▷ selecting right interval
  else
    $\alpha_R = \alpha_M$; $\text{AIVD}_R = \text{AIVD}_M$ ▷ selecting left interval
  end if
  return BISECT-1($\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \beta, k$) ▷ zooming in on selected interval
end function

Function DISPLACE-1 attempts to displace the midpoint α_M previously calculated at some stage in BISECT-1, in a way that its associated AIVD would fall outside the empirical uncertainty range. This function has all the arguments of BISECT-1 and α_M as an additional one, which is a real number belonging to $[0, 1]$. It returns the same type of information as FIT-1. The method first computes a "secondary" midpoint α'_M to the left of α_M and its corresponding AIVD'_M value. If the resolution limit $\delta\alpha$ is not reached and AIVD'_M falls outside the AIVD_{emp} range, BISECT-1 is applied further to the $[\alpha'_M, \alpha_R]$ interval (lines 2-11). Otherwise, the analogous procedure is applied on the right side (12-21). If the procedure fails to provide a convenient, secondary midpoint on either side, the fitting is considered complete with the current $[\alpha_L, \alpha_R]$ interval and the $\alpha_{\text{fit}}, \alpha_{\text{err}}$ estimates made like in BISECT-1 (lines 22-23).

1: **function** DISPLACE-1($\alpha_L, \alpha_R, \alpha_M, \text{AIVD}_L, \text{AIVD}_R, \beta, k$)
2:   $\alpha'_M = \text{MIDDLE}(\alpha_L, \alpha_M, \delta\alpha)$ ▷ trying displacement to the left on the α grid
3:   **if** DISTINCT($\alpha'_M, \alpha_L, \alpha_M$) **then**
4:     $\text{AIVD}'_M = \langle\text{AIVD}\rangle^k_{\alpha'_M,\beta}$ ▷ analytic calculations (see main text)
5:     **if** $\neg$MATCH-1($\text{AIVD}'_M, \text{AIVD}_{\text{emp}}$) **then**
6:       **if** ORD-1($\text{AIVD}_L, \text{AIVD}_R$) = ORD-1($\text{AIVD}'_M, \text{AIVD}_{\text{emp}}$) **then**
7:         $\alpha_L = \alpha'_M$; $\text{AIVD}_L = \text{AIVD}'_M$
8:         **return** BISECT-1($\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \beta, k$) ▷ zooming in on corrected interval
9:       **end if**
10:     **end if**
11:   **end if**
12:   $\alpha'_M = \text{MIDDLE}(\alpha_M, \alpha_R, \delta\alpha)$ ▷ trying displacement to the right on the α grid
13:   **if** DISTINCT($\alpha'_M, \alpha_M, \alpha_R$) **then**
14:     $\text{AIVD}'_M = \langle\text{AIVD}\rangle^k_{\alpha'_M,\beta}$ ▷ analytic calculations (see main text)
15:     **if** $\neg$MATCH-1($\text{AIVD}'_M, \text{AIVD}_{\text{emp}}$) **then**
16:       **if** ORD-1($\text{AIVD}_L, \text{AIVD}_R$) $\neq$ ORD-1($\text{AIVD}'_M, \text{AIVD}_{\text{emp}}$) **then**
17:         $\alpha_R = \alpha'_M$; $\text{AIVD}_R = \text{AIVD}'_M$
18:         **return** BISECT-1($\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \beta, k$) ▷ zooming in on corrected interval
19:       **end if**
20:     **end if**
21:   **end if**
22:   $(\alpha_{\text{fit}}, \alpha_{\text{err}}) = \text{INTERNFITLIN-1}(\alpha_L, \alpha_R, \text{AIVD}_L, \text{AIVD}_R, \text{AIVD}_{\text{emp}})$
23:   **return** $(\alpha_L, \alpha_R, \alpha_{\text{fit}}, \alpha_{\text{err}})$ ▷ fitting complete
24: **end function**

B. Second level fitting

This section presents the algorithm part concerned with the second fitting level. Each of the three functions of the first fitting level (Sec. A) has a correspondent here:

Fit-2, Bisect-2, Displace-2, all of them returning the same type of information [4], each of them having a similar, structure, purpose and role to the correspondent within the first fitting level. Additionally, this section presents the pseudocode for a fourth function, NumSIVD, which carries out the numerical SIVD calculations. In addition to the two constants introduced at the first level, the second level pseudocode invokes two other constants, which are also assumed to be known a-priori and available for use anywhere in these four functions. First, $\delta\beta$ is the desired resolution in the β parameter, which is here set to the inverse of the number of features: $\delta\beta = \frac{1}{F}$. Second, $\mathrm{SIVD}_{\mathrm{emp}}$ is the SIVD uncertainty range for the empirical data.

In relation to the first three functions, the descriptions below attempt to mostly emphasize the elements that come in addition with respect to their first-level correspondents. Some of these elements have a repetitive nature and are worth explaining before moving to the specific description of each function. First, the (generic) $\bar{\beta}_{\mathrm{X}}$ notation (where "X" can stand for "L", "R" or "M") denotes the (generic) "composite fitting information" $\bar{\beta}_{\mathrm{X}} = (\beta, \alpha_L, \alpha_R, \alpha_{\mathrm{fit}}, \alpha_{\mathrm{err}})_{\mathrm{X}}$, which is a 5-tuple consisting of a β value together with the associated four values returned by a (generic) call Fit-1(β, k) for that specific β and some arbitrary k. Second, whenever an "$\mathrm{SIVD}_{\mathrm{X}}$" variable appears in the first three functions (where "X" is again a generic label), except for $\mathrm{SIVD}_{\mathrm{emp}}$, it actually denotes the (generic) "composite SIVD information" $\mathrm{SIVD}_{\mathrm{X}} = ((\mathrm{SIVD}_L^{\mathrm{fit}}, \mathrm{SIVD}_L^{\mathrm{err}}), (\mathrm{SIVD}_R^{\mathrm{fit}}, \mathrm{SIVD}_R^{\mathrm{err}}))_{\mathrm{X}}$, which is a pair of pairs of real numbers, each inner pair corresponding to a model SIVD uncertainty range associated to one margin of an α interval returned by a call to Fit-1, while both inner pairs have the same β. This schematically reads:

$$(\beta, \alpha_L) \to (\mathrm{SIVD}_L^{\mathrm{fit}}, \mathrm{SIVD}_L^{\mathrm{err}}),$$
$$(\beta, \alpha_R) \to (\mathrm{SIVD}_R^{\mathrm{fit}}, \mathrm{SIVD}_R^{\mathrm{err}}),$$

Third, any (generic) call NumSIVD$(\bar{\beta}, k)$ is necessarily preceded by an associated (generic) call Fit-1(β, k) and by an associated (generic) variable compression $\bar{\beta} = (\beta, \alpha_L, \alpha_R, \alpha_{\mathrm{fit}}, \alpha_{\mathrm{err}})$, the last two being needed for producing the composite fitting information $\bar{\beta}$. Fourth, whenever a piece of composite SIVD information appears in a call to Ord-2 or Match-2, it is accompanied by an associated piece of composite fitting information, which allows for the mean, statistical error and systematic error of in the model SIVD to be all reconstructed within, for a given combination of β and k.

1: **function** Fit-2(k)
2: $(\beta_L, \beta_R) = \text{Init-2}(\delta\beta, k, \mathrm{AIVD}_{\mathrm{emp}})$ ▷ initializing the β-interval
3: **if** $\beta_L < \beta_R$ **then**
4: $(\alpha_L^L, \alpha_L^R, \alpha_L^{\mathrm{fit}}, \alpha_L^{\mathrm{err}}) = \text{Fit-1}(\beta_L, k)$; $(\alpha_R^L, \alpha_R^R, \alpha_R^{\mathrm{fit}}, \alpha_R^{\mathrm{err}}) = \text{Fit-1}(\beta_R, k)$
5: $\bar{\beta}_L = (\beta_L, \alpha_L^L, \alpha_L^R, \alpha_L^{\mathrm{fit}}, \alpha_L^{\mathrm{err}})$; $\bar{\beta}_R = (\beta_R, \alpha_R^L, \alpha_R^R, \alpha_R^{\mathrm{fit}}, \alpha_R^{\mathrm{err}})$
6: $\mathrm{SIVD}_L = \text{NumSIVD}(\bar{\beta}_L, k)$; $\mathrm{SIVD}_R = \text{NumSIVD}(\bar{\beta}_R, k)$ ▷ numeric calculations
7: **return** Bisect-2$(\bar{\beta}_L, \bar{\beta}_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k)$
8: **end if**
9: **return FittingImpossibleError**
10: **end function**

Function Fit-2 acts as an interface for the second-level fitting, which consists of tuning the β parameter, for a given value of k, such that the SIVD quantity matches the empirical value, relying on an underlying tuning of the α parameter in terms of the AIVD quantity (using Fit-1). Here, k is a strictly positive, integer number. The method returns the composite fitting information associated to the left ($\bar{\beta}_L$) and right ($\bar{\beta}_R$) margins of the tightest β interval found, together with the estimated β match within this interval (β_{fit}) and its associated error (β_{err}). It assumes that the empirical SIVD can actually be uniquely matched by varying β and α, for the given value of k. After checking that there exists a meaningful $[\beta_L, \beta_R]$ interval for which the first-level fitting is possible (lines 2,3), the method conducts the numeric SIVD calculations on both sides of the interval (line 6), preceded, on each side, by the first level fitting and the decompression (lines 4,5, as explained above), in order to finally pass the task to Bisect-2.

Function Bisect-2 is another recursive implementation of the bisection method, which sequentially narrows down the $[\beta_L, \beta_R]$ interval, such that at each stage the empirical SIVD is contained. Here, $\bar{\beta}_L, \bar{\beta}_R$ are 5-tuples of

[4] The type of information returned by the three functions at a second-level fitting is different than that of the three functions at the first-level fitting, and actually more complex.

real numbers encoding the left and right pieces of composite fitting information, $\mathrm{SIVD}_L, \mathrm{SIVD}_R$ are the pairs of pairs of real numbers encoding the left-β and right-β pieces of composite SIVD information, while k is of the same type as in Fit-2. It returns the same type of information as Fit-2. Like Bisect-1, the function consists of a part concerned with convergence (lines 4-7), a part concerned with the jump to Displace-2 (lines 11-13) and a part concerned with choosing between the left and right β subintervals and with zooming in on the chosen one (lines 14-19). Note the additional statements concerned with decompressing the composite fitting information (line 2) and with preparing the numeric SIVD calculations at the midpoint (lines 8-9).

1: **function** Bisect-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k$)
2: $\quad(\beta_L, \alpha_L^L, \alpha_L^R, \alpha_L^{\mathrm{fit}}, \alpha_L^{\mathrm{err}}) = \bar\beta_L$; $(\beta_R, \alpha_R^L, \alpha_R^R, \alpha_R^{\mathrm{fit}}, \alpha_R^{\mathrm{err}}) = \bar\beta_R$
3: $\quad\beta_M = $ Middle($\beta_L, \beta_R, \delta\beta$)
4: $\quad$**if** $\neg$Distinct($\beta_M, \beta_L, \beta_R$) **then**
5: $\qquad(\beta_{\mathrm{fit}}, \beta_{\mathrm{err}}) = $ InternFitLin-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, \mathrm{SIVD}_{\mathrm{emp}}$)
6: $\qquad$**return** $(\bar\beta_L, \bar\beta_R, \beta_{\mathrm{fit}}, \beta_{\mathrm{err}})$
7: $\quad$**end if**
8: $\quad(\alpha_M^L, \alpha_M^R, \alpha_M^{\mathrm{fit}}, \alpha_M^{\mathrm{err}}) = $ Fit-1(β_M, k)
9: $\quad\bar\beta_M = (\beta_M, \alpha_M^L, \alpha_M^R, \alpha_M^{\mathrm{fit}}, \alpha_M^{\mathrm{err}})$
10: $\quad\mathrm{SIVD}_M = $ NumSIVD($\bar\beta_M, k$) $\triangleright$ numeric calculations
11: $\quad$**if** Match-2($\bar\beta_M, \mathrm{SIVD}_M, \mathrm{SIVD}_{\mathrm{emp}}$) **then**
12: $\qquad$**return** Displace-2($\bar\beta_L, \bar\beta_R, \bar\beta_M, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k$)
13: $\quad$**end if**
14: $\quad$**if** Ord-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R$) = Ord-2($\bar\beta_M, \mathrm{SIVD}_M, \mathrm{SIVD}_{\mathrm{emp}}$) **then**
15: $\qquad\bar\beta_L = \bar\beta_M$; $\mathrm{SIVD}_L = \mathrm{SIVD}_M$ $\triangleright$ selecting right interval
16: $\quad$**else**
17: $\qquad\bar\beta_R = \bar\beta_M$; $\mathrm{SIVD}_R = \mathrm{SIVD}_M$ $\triangleright$ selecting left interval
18: $\quad$**end if**
19: $\quad$**return** Bisect-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k$) $\triangleright$ zooming in on selected interval
20: **end function**

Function Displace-2 attempts to displace the midpoint β_M previously calculated at some stage in Bisect-2, in a way that its associated SIVD uncertainty range does not overlap with the empirical one. This function has all the arguments of Bisect-1 and $\bar\beta_M$ as an additional one, which is a 5-tuple of real numbers encoding the midpoint composite fitting information. It returns the same type of information as Fit-2. Like Displace-1, the function consists of a part that attempts a displacement to the left (lines 3-14), one that attempts a displacement to the right (lines 15-26) and one that takes care of the convergence (lines 27-28). Note the additional statements concerned with decompressing the composite fitting information (line 2) and with preparing the numeric SIVD calculations for the left/right secondary midpoint (lines 5-6/17-18).

1: **function** Displace-2($\bar\beta_L, \bar\beta_R, \bar\beta_M, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k$)
2: $\quad(\beta_L, \alpha_L^L, \alpha_L^R, \alpha_L^{\mathrm{fit}}, \alpha_L^{\mathrm{err}}) = \bar\beta_L$; $(\beta_R, \alpha_R^L, \alpha_R^R, \alpha_R^{\mathrm{fit}}, \alpha_R^{\mathrm{err}}) = \bar\beta_R$; $(\beta_M, \alpha_M^L, \alpha_M^R, \alpha_M^{\mathrm{fit}}, \alpha_M^{\mathrm{err}}) = \bar\beta_M$
3: $\quad\beta'_M = $ Middle($\beta_L, \beta_M, \delta\beta$) $\triangleright$ trying displacement to the left
4: $\quad$**if** Distinct($\beta'_M, \beta_L, \beta_M$) **then**
5: $\qquad(\dot\alpha_M^L, \dot\alpha_M^R, \dot\alpha_M^{\mathrm{fit}}, \dot\alpha_M^{\mathrm{err}}) = $ Fit-1(β'_M, k)
6: $\qquad\bar\beta'_M = (\beta'_M, \dot\alpha_M^L, \dot\alpha_M^R, \dot\alpha_M^{\mathrm{fit}}, \dot\alpha_M^{\mathrm{err}})$
7: $\qquad\mathrm{SIVD}'_M = $ NumSIVD($\bar\beta'_M, k$) $\triangleright$ numeric calculations
8: $\qquad$**if** $\neg$Match-2($\bar\beta'_M, \mathrm{SIVD}'_M, \mathrm{SIVD}_{\mathrm{emp}}$) **then**
9: $\qquad\quad$**if** Ord-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R$) = Ord-2($\bar\beta'_M, \mathrm{SIVD}'_M, \mathrm{SIVD}_{\mathrm{emp}}$) **then**
10: $\qquad\qquad\bar\beta_L = \bar\beta'_M$; $\mathrm{SIVD}_L = \mathrm{SIVD}'_M$
11: $\qquad\qquad$**return** Bisect-2($\bar\beta_L, \bar\beta_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, k$) $\triangleright$ zooming in on corrected interval
12: $\qquad\quad$**end if**
13: $\qquad$**end if**
14: $\quad$**end if**
15: $\quad\beta'_M = $ Middle($\beta_M, \beta_R, \delta\beta$) $\triangleright$ trying displacement to the right
16: $\quad$**if** Distinct($\beta'_M, \beta_M, \beta_R$) **then**

$(\dot\alpha_M^L, \dot\alpha_M^R, \dot\alpha_M^{\text{fit}}, \dot\alpha_M^{\text{err}}) = \textsc{Fit-1}(\beta'_M, k)$
$\bar\beta'_M = (\beta'_M, \dot\alpha_M^L, \dot\alpha_M^R, \dot\alpha_M^{\text{fit}}, \dot\alpha_M^{\text{err}})$
$\text{SIVD}'_M = \textsc{NumSIVD}(\bar\beta'_M, k)$ ▷ numeric calculations
if $\neg\textsc{Match-2}(\bar\beta'_M, \text{SIVD}'_M, \text{SIVD}_{\text{emp}})$ **then**
  if $\textsc{Ord-2}(\bar\beta_L, \bar\beta_R, \text{SIVD}_L, \text{SIVD}_R) \neq \textsc{Ord-2}(\bar\beta'_M, \text{SIVD}'_M, \text{SIVD}_{\text{emp}})$ **then**
    $\bar\beta_R = \bar\beta'_M$; $\text{SIVD}_R = \text{SIVD}'_M$
    return $\textsc{Bisect-2}(\bar\beta_L, \bar\beta_R, \text{SIVD}_L, \text{SIVD}_R, k)$ ▷ zooming in on corrected interval
  end if
end if
end if
$(\beta_{\text{fit}}, \beta_{\text{err}}) = \textsc{InternFitLin-2}(\bar\beta_L, \bar\beta_R, \text{SIVD}_L, \text{SIVD}_R, \text{SIVD}_{\text{emp}})$
return $(\bar\beta_L, \bar\beta_R, \beta_{\text{fit}}, \beta_{\text{err}})$
end function

Function NumSIVD numerically generates a piece of composite SIVD information with a precision that is as high as possible. Here, $\bar\beta$ is a 5-tuple of real numbers encoding a composite fitting information, while k is a positive integer number. One sequence of SIVD values is numerically generated (lines 4 and 11) for each of the two margins of the α interval (contained in $\bar\beta$), for the given β (also contained in $\bar\beta$) and the given k. An uncertainty range is obtained from each of the two sequences (lines 5 and 13). These two uncertainty ranges are used together with the information in $\bar\beta$ to produce estimates for an average, a statistical error and a systematic error that are $\bar\beta$-specific rather than (α, β)-specific (lines 6,7 and 14,15). The number of SIVD values in the two sequences is increased and the calculations are repeated as long as the condition in line 9 remains true, namely as long as: the statistical error is higher than the systematic error, the desired separation between the model and empirical (statistical) uncertainty ranges is not reached and the maximal SIVD sequence length is not reached. The desired separation and the SIVD sequence length are controlled via variables s and n, initialized in line 2 – the initial values of these variables, as well as the upper bound on the latter are hard-coded, as visible in the pseudocode, and have been decided after some experimentation with NumSIVD, but they are not essential for the actual outcome. Also note the decompression of the composite fitting information (line 3) and the decompression of SIVD uncertainty ranges (lines 8 and 16).

1: **function** $\textsc{NumSIVD}(\bar\beta, k)$
2:   $n = 20$; $s = 5$ ▷ initial number of realizations and desired separation
3:   $(\beta, \alpha_L, \alpha_R, \alpha_{\text{fit}}, \alpha_{\text{err}}) = \bar\beta$
4:   $\text{SIVD}_L^{\text{seq}} = \textsc{GenSeqSIVD}(\alpha_L, \beta, k, n)$; $\text{SIVD}_R^{\text{seq}} = \textsc{GenSeqSIVD}(\alpha_R, \beta, k, n)$
5:   $\text{SIVD}_L = \textsc{CompAvgErr}(\text{SIVD}_L^{\text{seq}})$; $\text{SIVD}_R = \textsc{CompAvgErr}(\text{SIVD}_R^{\text{seq}})$
6:   $\text{SIVD} = \textsc{Interpol}(\alpha_L, \alpha_R, \alpha_{\text{fit}}, \text{SIVD}_L, \text{SIVD}_R)$
7:   $\text{SIVD}_{\text{syst}} = \textsc{CompSystErr}(\alpha_L, \alpha_R, \alpha_{\text{err}}, \text{SIVD}_L, \text{SIVD}_R)$
8:   $(\text{SIVD}_{\text{avg}}, \text{SIVD}_{\text{stat}}) = \text{SIVD}$; $(\text{SIVD}_{\text{emp}}^{\text{avg}}, \text{SIVD}_{\text{emp}}^{\text{stat}}) = \text{SIVD}_{\text{emp}}$
9:   **while** $\text{SIVD}_{\text{stat}} > \text{SIVD}_{\text{syst}} \wedge (\text{SIVD}_{\text{stat}} + \text{SIVD}_{\text{emp}}^{\text{stat}} > |\text{SIVD}_{\text{emp}}^{\text{avg}} - \text{SIVD}_{\text{avg}}|/s) \wedge n < 350$ **do**
10:     $n = 2 \cdot n$
11:     $\text{SIVD}_L^{\text{tmpSeq}} = \textsc{GenSeqSIVD}(\alpha_L, \beta, k, n)$; $\text{SIVD}_R^{\text{tmpSeq}} = \textsc{GenSeqSIVD}(\alpha_R, \beta, k, n)$
12:     $\text{SIVD}_L^{\text{seq}} = \textsc{Merge}(\text{SIVD}_L^{\text{seq}}, \text{SIVD}_L^{\text{tmpSeq}})$; $\text{SIVD}_R^{\text{seq}} = \textsc{Merge}(\text{SIVD}_R^{\text{seq}}, \text{SIVD}_R^{\text{tmpSeq}})$
13:     $\text{SIVD}_L = \textsc{CompAvgErr}(\text{SIVD}_L^{\text{seq}})$; $\text{SIVD}_R = \textsc{CompAvgErr}(\text{SIVD}_R^{\text{seq}})$
14:     $\text{SIVD} = \textsc{Interpol}(\alpha_L, \alpha_R, \alpha_{\text{fit}}, \text{SIVD}_L, \text{SIVD}_R)$
15:     $\text{SIVD}_{\text{syst}} = \textsc{CompSystErr}(\alpha_L, \alpha_R, \alpha_{\text{err}}, \text{SIVD}_L, \text{SIVD}_R)$
16:     $(\text{SIVD}_{\text{avg}}, \text{SIVD}_{\text{stat}}) = \text{SIVD}$
17:   **end while**
18:   **return** $(\text{SIVD}_L, \text{SIVD}_R)$
19: **end function**

C. Used functions

This section describes functions that are used by the pseudocode in sections A or B but are not described there. The following is a list of functions for which the pseudocode is also provided, following each text description.

Function InterfitLin-1 fine-tunes the α parameter such that $\mathrm{AIVD}_{\mathrm{emp}}$ is matched, relying on a linear approximation of the model AIVD as a function of α within the (α_L, α_R) interval, using the boundary values AIVD_L and AIVD_R. Its arguments are of the same type as those of InterFitLin (described below), except that AIVD_L and AIVD_R are real numbers rather than uncertainty ranges. The output structure is entirely the same as that of InterFitLin. It is essentially a first-level fitting interface for InternFitLin, which is called after specifying that the errors associated to AIVD_L and AIVD_R are zero.

function InternFitLin-1$(\alpha_L, \alpha_R, \mathrm{AIVD}_L, \mathrm{AIVD}_R, \mathrm{AIVD}_{\mathrm{emp}})$
  $\mathrm{AIVD}'_L = (\mathrm{AIVD}_L, 0)$; $\mathrm{AIVD}'_R = (\mathrm{AIVD}_R, 0)$
  return InternFitLin$(\alpha_L, \alpha_R, \mathrm{AIVD}'_L, \mathrm{AIVD}'_R, \mathrm{AIVD}_{\mathrm{emp}})$
end function

Function InterfitLin-2 fine-tunes the β parameter such that $\mathrm{SIVD}_{\mathrm{emp}}$ is matched, relying on a linear approximation of the model SIVD as a function of β within the $[\beta_L, \beta_R]$ interval, using the boundary information stored in SIVD_L and SIVD_R. Its arguments are of the same type as those of InterFitLin (described below), except that $\bar{\beta}_L$ and $\bar{\beta}_R$ are 5-tuples or real numbers rather than real numbers and SIVD_L and SIVD_R are pieces composite SIVD information rather than uncertainty ranges. The output structure is entirely the same as that of InterFitLin. It is essentially a second-level fitting interface for InternFitLin, which is called after carrying out the following two operations: computing the mean, statistical error and systematic error on each of the two margins of the β interval, using the right combination of composite fitting information and composite SIVD information (lines 2,3); compressing information into an SIVD uncertainty range for each of the two margins, after choosing the highest among the two errors for each margin.

1: **function** InternFitLin-2$(\bar{\beta}_L, \bar{\beta}_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R, \mathrm{SIVD}_{\mathrm{emp}})$
2: $(\mathrm{SIVD}_L^{\mathrm{avg}}, \mathrm{SIVD}_L^{\mathrm{stat}}, \mathrm{SIVD}_L^{\mathrm{syst}}) = \textsc{MeanStatSyst}(\bar{\beta}_L, \mathrm{SIVD}_L)$
3: $(\mathrm{SIVD}_R^{\mathrm{avg}}, \mathrm{SIVD}_R^{\mathrm{stat}}, \mathrm{SIVD}_R^{\mathrm{syst}}) = \textsc{MeanStatSyst}(\bar{\beta}_R, \mathrm{SIVD}_R)$
4: $\mathrm{SIVD}'_L = (\mathrm{SIVD}_L^{\mathrm{avg}}, \textsc{max}(\mathrm{SIVD}_L^{\mathrm{stat}}, \mathrm{SIVD}_L^{\mathrm{syst}}))$
5: $\mathrm{SIVD}'_R = (\mathrm{SIVD}_R^{\mathrm{avg}}, \textsc{max}(\mathrm{SIVD}_R^{\mathrm{stat}}, \mathrm{SIVD}_R^{\mathrm{syst}}))$
6: **return** InternFitLin$(\beta_L, \beta_R, \mathrm{SIVD}'_L, \mathrm{SIVD}'_R, \mathrm{SIVD}_{\mathrm{emp}})$
7: **end function**

Function Match-1 checks whether AIVD (real value) falls within the uncertainty range specified by $\mathrm{AIVD}_{\mathrm{emp}}$. It acts as an interface for Match (described below) within the first-level fitting scheme.

function Match-1$(\mathrm{AIVD}, \mathrm{AIVD}_{\mathrm{emp}})$
  $\mathrm{AIVD}' = (\mathrm{AIVD}, 0)$
  return Match$(\mathrm{AIVD}', \mathrm{AIVD}_{\mathrm{emp}})$
end function

Function Match-2 checks whether there is an overlap between the model SIVD uncertainty range obtained from $\bar{\beta}$ (composite fitting information) and SIVD (composite SIVD information) and the empirical one encoded by $\mathrm{SIVD}_{\mathrm{emp}}$. It acts as an interface for Match (described below) within the second-level fitting scheme.

function Match-2$(\bar{\beta}, \mathrm{SIVD}, \mathrm{SIVD}_{\mathrm{emp}})$
  $(\mathrm{SIVD}_{\mathrm{avg}}, \mathrm{SIVD}_{\mathrm{stat}}, \mathrm{SIVD}_{\mathrm{syst}}) = \textsc{MeanStatSyst}(\bar{\beta}, \mathrm{SIVD})$
  $\mathrm{SIVD}' = (\mathrm{SIVD}_{\mathrm{avg}}, \textsc{max}(\mathrm{SIVD}_{\mathrm{stat}}, \mathrm{SIVD}_{\mathrm{syst}}))$
  return Match$(\mathrm{SIVD}', \mathrm{SIVD}_{\mathrm{emp}})$

end function

Function Ord-1 (first version) checks whether AIVD_L (real value) is smaller than AIVD_R (real value), acting as an interface for Ord within the first-level fitting scheme.

function Ord-1($\mathrm{AIVD}_L, \mathrm{AIVD}_R$)
  return Ord($\mathrm{AIVD}_L, \mathrm{AIVD}_R$)
end function

Function Ord-1 (second version) checks whether AIVD (real value) is smaller than the average stored in $\mathrm{AIVD}_{\mathrm{emp}}$ (uncertainty range), acting as an interface for Ord within the first-level fitting scheme.

function Ord-1($\mathrm{AIVD}, \mathrm{AIVD}_{\mathrm{emp}}$)
  $(\mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{avg}}, \mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{err}}) = \mathrm{AIVD}_{\mathrm{emp}}$
  return Ord($\mathrm{AIVD}, \mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{avg}}$)
end function

Function Ord-2 (first version) checks whether the average stored in the SIVD uncertainty range obtained from $\bar{\beta}_L$ (composite fitting information) and SIVD_L (composite SIVD information) is smaller than the average stored in that obtained from $\bar{\beta}_R$ (composite fitting information) and SIVD_R (composite SIVD information), acting as an interface for Ord within the second-level fitting scheme.

function Ord-2($\bar{\beta}_L, \bar{\beta}_R, \mathrm{SIVD}_L, \mathrm{SIVD}_R$)
  $(\mathrm{SIVD}_L^{\mathrm{avg}}, \mathrm{SIVD}_L^{\mathrm{stat}}, \mathrm{SIVD}_L^{\mathrm{syst}}) =$ MeanStatSyst($\bar{\beta}_L, \mathrm{SIVD}_L$)
  $(\mathrm{SIVD}_R^{\mathrm{avg}}, \mathrm{SIVD}_R^{\mathrm{stat}}, \mathrm{SIVD}_R^{\mathrm{syst}}) =$ MeanStatSyst($\bar{\beta}_R, \mathrm{SIVD}_R$)
  return Ord($\mathrm{SIVD}_L^{\mathrm{avg}}, \mathrm{SIVD}_R^{\mathrm{avg}}$)
end function

Function Ord-2 (second version) checks whether the average stored in the SIVD uncertainty range obtained from $\bar{\beta}$ (composite fitting information) and SIVD (composite SIVD information) is smaller than the average stored $\mathrm{SIVD}_{\mathrm{emp}}$, acting as an interface for Ord within the second-level fitting scheme.

function Ord-2($\bar{\beta}, \mathrm{SIVD}, \mathrm{SIVD}_{\mathrm{emp}}$)
  $(\mathrm{SIVD}_{\mathrm{avg}}, \mathrm{SIVD}_{\mathrm{stat}}, \mathrm{SIVD}_{\mathrm{syst}}) =$ MeanStatSyst($\bar{\beta}, \mathrm{SIVD}$)
  $(\mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{avg}}, \mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{err}}) = \mathrm{AIVD}_{\mathrm{emp}}$
  return Ord($\mathrm{SIVD}_{\mathrm{avg}}, \mathrm{AIVD}_{\mathrm{emp}}^{\mathrm{avg}}$)
end function

Function MeanStatSyst estimates a mean, a statistical error and a systematic error from a piece of composite fitting information and an associated piece of composite SIVD information, which are the two arguments of the function. It returns the 3-tuple comprising of the three computed real numbers. Note the decompression of composite fitting information (line 2) and the decompression of composite SIVD information (line 3).

1: **function** MeanStatSyst($\bar{\beta}, \mathrm{SIVD}$)

$(\beta, \alpha_L, \alpha_R, \alpha_{\text{fit}}, \alpha_{\text{err}}) = \bar{\beta}$
$(\text{SIVD}_L, \text{SIVD}_R) = \text{SIVD}$
$\text{SIVD}' = \textsc{Interpol}(\alpha_L, \alpha_R, \alpha_{\text{fit}}, \text{SIVD}_L, \text{SIVD}_R)$
$\text{SIVD}_{\text{syst}} = \textsc{CompSystErr}(\alpha_L, \alpha_R, \alpha_{\text{err}}, \text{SIVD}_L, \text{SIVD}_R)$
$(\text{SIVD}_{\text{avg}}, \text{SIVD}_{\text{stat}}) = \text{SIVD}'$
return $(\text{SIVD}_{\text{avg}}, \text{SIVD}_{\text{stat}}, \text{SIVD}_{\text{syst}})$
end function

The following is a list of functions for which only text explanations are provided in schematic way, sometimes accompanied by figures.

- $\textsc{Init-1}(\delta\alpha)$:
  - gives the left and right boundaries of the largest possible interval for which the α parameter is compatible with the stochastic model in use, given the grid spacing $\delta\alpha$
  - input: δ is a real number
  - in practice it returns $(\delta\alpha, 1-\delta\alpha)$ regardless of whether PG or MPG is used
- $\textsc{Init-2}(\delta\beta, k, \text{AIVD}_{\text{emp}})$:
  - gives the left and right boundaries of the largest possible interval, if any, for which the β parameter allows for the (first level) fitting of $\text{AIVD}(\alpha)$ to successfully take place, given the grid spacing $\delta\beta$
  - input: $\delta\beta$ is a real number, k is a positive integer and AIVD_{emp} is an uncertainty range
  - assumes that there exists at most one β interval $[\beta_L, \beta_R]$ for which there exists an α such that $\langle\text{AIVD}\rangle^k_{\alpha,\beta} = \text{AIVD}_{\text{emp}}$ is satisfied
  - starts from the largest interval allowed by the model and independently adjusts each of the two boundaries via a branching algorithm, until the desired interval is reached
  - returns two (incompatible) boundaries $\beta_L > \beta_R$ if such an interval does not exist
- $\textsc{Middle}(l, r, \delta)$:
  - computes the value closest to the average between l and r, on a grid of spacing δ
  - input: l, r, δ are all real numbers
  - assumes that the interval length $l - r$ is equal to an integer times δ
- $\textsc{Distinct}(m, l, r)$:
  - checks whether m is different than both l and r
  - input: m, l, r are all real numbers constrained constrained to a grid of constant spacing
- $\textsc{InternFitLin}(p_L, p_R, O_L, O_R, O_{\text{emp}})$:
  - adjusts a parameter p such that an observable O attains a value compatible with the empirical in O_{emp} interval, assuming that O is a linear function of p within the $[p_L, p_R]$ interval
  - input: p_L, p_R are real numbers, encoding the left and right boundaries of the interval; O_L, O_R are mean-error pairs of real numbers encoding the theoretical uncertainty ranges of the observable for the left and for the right boundaries; O_{emp} is a mean-error pair of real numbers encoding the empirical uncertainty range
  - returns the value and associated error of the p parameter resulting from this fitting process $(p_{\text{fit}}, p_{\text{err}})$, computed based on geometrical considerations, in the manner illustrated in Fig. 1(a)
  - p_{fit} is calculated first by intersecting the theoretical line with the empirical one, disregarding all errors; then, p_{err} is calculated by assuming that the theoretical error is constant within the $[p_L, p_R]$ interval, with value given by interpolating the errors contained by O_L and O_R at p_{fit}
  - p_{err} takes its origin both in the the empirical error as well as in the theoretical error, but also depends on the slope resulting from the linear approximation

- MATCH(r_1, r_2):
  - checks whether there is an overlap between the uncertainty ranges encoded by r_1 and r_2
  - input: r_1, r_2 are mean-error pairs of real numbers
- ORD(v_L, v_R):
  - checks whether the condition $v_L < v_R$ is satisfied
  - input: v_L, v_R are real numbers
  - assumes that $v_L \neq v_R$
- GENSEQSIVD(α, β, k, n)
  - numerically generates a sequence of n SIVD values according to the respective stochastic model, subject to parameter values indicated by k, α, β
  - input: α, β are real numbers, while k, n are positive integers
- MERGE($\mathrm{SIVD}_1^{\mathrm{seq}}, \mathrm{SIVD}_2^{\mathrm{seq}}$)
  - merges two sequences of (real) SIVD values
  - input: $\mathrm{SIVD}_1^{\mathrm{seq}}, \mathrm{SIVD}_2^{\mathrm{seq}}$ are both sequences of (real) SIVD values
- COMPAVGERR($\mathrm{SIVD}^{\mathrm{seq}}$)
  - computes the mean and standard error of the mean from $\mathrm{SIVD}^{\mathrm{seq}}$
  - input: $\mathrm{SIVD}^{\mathrm{seq}}$ is a sequence of real SIVD values
- INTERPOL($\alpha_L, \alpha_R, \alpha_{\mathrm{fit}}, \mathrm{SIVD}_L, \mathrm{SIVD}_R$)
  - estimates the mean and error in SIVD corresponding to α_{fit} based on the values attained for α_L
  - input: $\alpha_L, \alpha_R, \alpha_{\mathrm{fit}}$ are real numbers, while $\mathrm{SIVD}_L, \mathrm{SIVD}_R$ are mean-error pairs of real numbers
  - uses on a linear interpolation within the $[\alpha_L, \alpha_R]$ interval, separately for the mean and for the error
- COMPSYSTERR($\alpha_L, \alpha_R, \alpha_{\mathrm{err}}, \mathrm{SIVD}_L, \mathrm{SIVD}_R$)
  - estimates the systematic error $\mathrm{SIVD}^{\mathrm{syst}}$ of the SIVD quantity induced by the error α_{err} (associated to fitting the α parameter in terms of the AIVD quantity), assuming that SIVD is a linear function of α within the $[\alpha_L, \alpha_R]$ interval
  - input: $\alpha_L, \alpha_R, \alpha_{\mathrm{err}}$ are real numbers while while $\mathrm{SIVD}_L, \mathrm{SIVD}_R$ are mean-error pairs of real numbers encoding the theoretical uncertainty ranges on the left and right boundaries
  - $\mathrm{SIVD}^{\mathrm{syst}}$ is computed based on geometrical considerations, in the manner illustrated in Fig. 1(b)

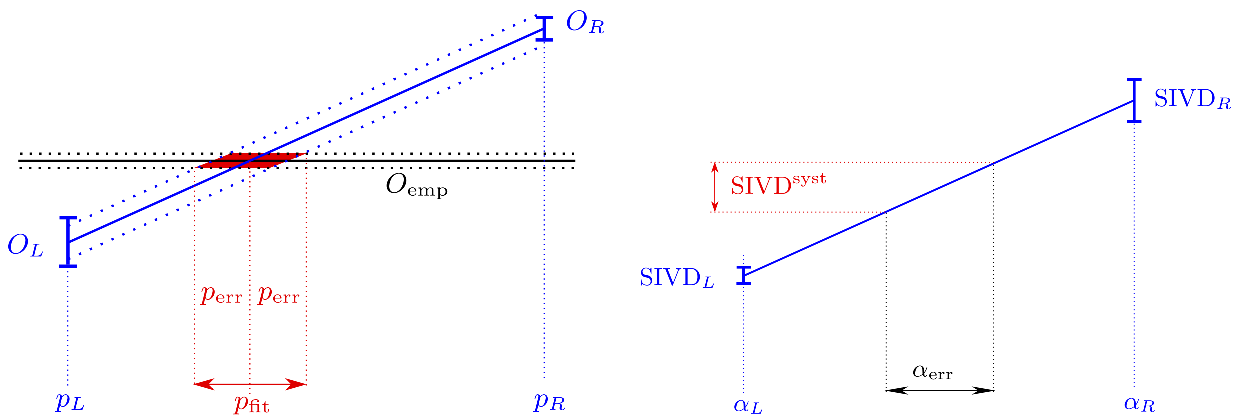
D. Algorithm usage

This section explain how the formalism presented throughout this document is effectively used for producing the results shown in the "Model fitting" and "Model outcomes" sections of the main text.

First, the formalism is used for producing the plots showing the SIVD(β) dependence ("Model fitting" section). For either PG or MPG, for a specific k value and a specific β on-grid value, the drawn model SIVD uncertainty range is obtained after the following computational steps:

1: $(\alpha_L, \alpha_R, \alpha_{\mathrm{fit}}, \alpha_{\mathrm{err}}) = \mathrm{FIT\text{-}1}(\beta, k)$ ▷ executing 1st-level fitting
2: $\bar{\beta} = (\beta, \alpha_L, \alpha_R, \alpha_{\mathrm{fit}}, \alpha_{\mathrm{err}})$ ▷ creating composite fitting information
3: $\mathrm{SIVD} = \mathrm{NUMSIVD}(\bar{\beta}, k)$ ▷ numeric SIVD calculations
4: $(\mathrm{SIVD}_{\mathrm{avg}}, \mathrm{SIVD}_{\mathrm{stat}}, \mathrm{SIVD}_{\mathrm{syst}}) = \mathrm{MEANSTATSYST}(\bar{\beta}, \mathrm{SIVD})$

which provides the values of the SIVD average $\mathrm{SIVD}_{\mathrm{avg}}$, the SIVD statistical error $\mathrm{SIVD}_{\mathrm{stat}}$ and the SIVD system-



(a) (b)

FIG. 1. Illustration of computation carried out by InterFitLin (a) and by CompSystErr (b), with the output quantities highlighted in red.

atic error $\mathrm{SIVD}_{\rm syst}$. One can then place a point at coordinates $(\beta, \mathrm{SIVD}_{\rm avg})$, within the respective k curve, with an error bar given by the maximum between $\mathrm{SIVD}_{\rm stat}$ and $\mathrm{SIVD}_{\rm syst}$.

Second, the formalism is used for providing the best-fitting, on-grid values for the α and β model parameters, which are used for generating sets of cultural vectors on which the LTCD-STCB analysis is applied ("Model outcomes" section). For either PG or MPG and for a specific k value, the following procedure is followed:

$(\bar\beta_L, \bar\beta_R, \beta_{\rm fit}, \beta_{\rm err}) = \text{Fit-2}(k)$ ▷ Executing 2nd-level fitting
$(\beta_L, \alpha_L^L, \alpha_L^R, \alpha_L^{\rm fit}, \alpha_L^{\rm err}) = \bar\beta_L$ ▷ Decompressing left-β composite fitting information
$(\beta_R, \alpha_R^L, \alpha_R^R, \alpha_R^{\rm fit}, \alpha_R^{\rm err}) = \bar\beta_R$ ▷ Decompressing right-β composite fitting information
if $\beta_{\rm fit} - \beta_L < \beta_R - \beta_{\rm fit}$ **then**
    $\beta = \beta_L$ ▷ choosing β_L, since it is closer to β
    if $\alpha_L^{\rm fit} - \alpha_L^L < \alpha_L^R - \alpha_L^{\rm fit}$ **then**
        $\alpha = \alpha_L^L$ ▷ choosing α_L^L, since it is closer to $\alpha_L^{\rm fit}$
    else
        $\alpha = \alpha_L^R$ ▷ choosing α_L^R, since it is closer to $\alpha_L^{\rm fit}$
    end if
else
    $\beta = \beta_R$ ▷ choosing β_R, since it is closer to β
    if $\alpha_R^{\rm fit} - \alpha_R^L < \alpha_R^R - \alpha_R^{\rm fit}$ **then**
        $\alpha = \alpha_R^L$ ▷ choosing α_R^L, since it is closer to $\alpha_R^{\rm fit}$
    else
        $\alpha = \alpha_R^R$ ▷ choosing α_R^R, since it is closer to $\alpha_R^{\rm fit}$
    end if
end if

which provides the best on-grid values for the (α, β) pair.