Root Tracking for Rate-Distortion: Approximating a Solution Curve with Higher Implicit Multivariate Derivatives

Shlomi Agmon[1*]

[1] School of Computer Science and Engineering,
The Hebrew University of Jerusalem, Jerusalem, Israel
Email: shlomi.agmon@mail.huji.ac.il

Abstract

The rate-distortion curve captures the fundamental tradeoff between compression length and resolution in lossy data compression. However, it conceals the underlying dynamics of optimal source encodings or *test channels*. We argue that these typically follow a piecewise smooth trajectory as the source information is compressed. These smooth dynamics are interrupted at *bifurcations*, where solutions change qualitatively. Sub-optimal test channels may collide or exchange optimality there, for example. There is typically a plethora of sub-optimal solutions, which stems from restrictions of the reproduction alphabet.

We devise a family of algorithms that exploits the underlying dynamics to track a given test channel along the rate-distortion curve. To that end, we express implicit derivatives at the roots of a non-linear operator by higher derivative tensors. Providing closed-form formulae for the derivative tensors of Blahut's algorithm thus yields implicit derivatives of arbitrary order at a given test channel, thereby approximating others in its vicinity. Finally, our understanding of bifurcations guarantees the optimality of the root being traced, under mild assumptions, while allowing us to detect when our assumptions fail.

Beyond the interest in rate distortion, this is an example of how understanding a problem's bifurcations can be translated to a numerical algorithm.

Keywords: Rate distortion theory, Bifurcation, Differential equations.

1 Introduction

The theory of lossy data compression was introduced in the seminal works of Shannon [1948, 1959], and has since found multiple applications beyond the obvious ones of communications and storage of information. Among others, clustering [Rose, 1998], perception [Blau and Michaeli, 2019, Sims, 2016], and it is intimately related to the information bottleneck principle [Tishby et al., 1999] in learning, [Gilad-Bachrach et al., 2003].

Formally, let $X \sim p_X$ and $\hat{X}$ be discrete i.i.d. random variables on finite *source* and *reproduction* alphabets, respectively denoted $\mathcal{X}$ and $\hat{\mathcal{X}}$. A *rate distortion* (RD) problem is defined by a

---

distortion measure $d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ and a source distribution $p_X(x)$, or $p(x)$ for short. One seeks [Shannon, 1948, 1959] the minimal rate $I(X; \hat{X}) := \mathbb{E}_{p(\hat{x}|x)p_X(x)} \log \frac{p(\hat{x}|x)}{p(\hat{x})}$ subject to a constraint D on the expected distortion. The optimal tradeoff between the information rate per message to the distortion is encoded by the rate-distortion function

$$R(D) := \min_{p(\hat{x}|x)} \left\{ I(X; \hat{X}) : \ \mathbb{E}_{p(\hat{x}|x)p_X(x)} [d(x, \hat{x})] \leq D \right\} . \tag{1.1}$$

Despite the interest in rate distortion, there are surprisingly few ways to calculate this tradeoff.

While the minimization problem (1.1) can be solved analytically in special cases, e.g., [Berger, 1971, 2.6] or [Cover and Thomas, 2006, 10.3], a solution is often obtained numerically by the iterative Blahut-Arimoto (BA) algorithm, due to Blahut [1972]. Using alternating minimizations, it converges to a test channel $p(\hat{x}|x)$ which achieves the minimum of (1.1), Csiszár [1974]. However, the BA algorithm suffers from *critical slowing down* near critical points, Agmon et al. [2021], points at which the number of symbols $\hat{x}$ required for optimal reproduction decreases. That is, there is a dramatic increase in the computational costs until convergence there. Further, one is often interested in the entire $R(D)$ curve. However, standard computation techniques solve (1.1) only at specified grid points. This yields isolated samples along the curve while making little to no use of previously computed solutions.

To alleviate computational costs, one could consider more efficient BA variants or its approximations thereof, such as the related [Yu, 2010, Matz and Duhamel, 2004, Sayir, 2000] for channel capacity or [Sutter et al., 2015] for constrained capacity. Alternatively, the choice of initial condition could be improved. For example, deterministic annealing [Rose et al., 1990] uses the solution at each grid-point as the initialization for the next, in analogy to annealing in statistical physics [Rose, 1998]. The algorithms we propose aim to improve upon this choice significantly.

In an attempt to tackle the above shortcomings of BA, we propose a new family of algorithms whose purpose is to follow the path of a known solution as some (scalar) control parameter is varied. This is especially appealing for RD problems, as a solution at the extremities of the curve is nearly trivial to obtain. For, the constant encoding to $\operatorname{argmin}_{\hat{x}} \mathbb{E}[d(X, \hat{x})]$ and $x \mapsto \operatorname{argmin}_{\hat{x}} d(x, \hat{x})$ are respectively optimal at zero rate or when the smallest distortion is desired. Unlike BA, our algorithm for RD provides a piecewise polynomial approximation of the path traversed by the distributions achieving (1.1), with uniform convergence guarantees outside a small vicinity of the critical points. Building on the work of Agmon et al. [2021], our algorithm does *not* suffer from an increased computational cost near critical points, but rather from a reduced accuracy there. Nevertheless, it admits a comprehensible tradeoff between accuracy and computational cost, permitting intelligible choices when high accuracy is desired. The computational cost of our algorithm is comparable to that of BA with reverse deterministic annealing, with the advantage of computing the entire curve of solutions rather than only solving on a grid.

Using the method of Lagrange multipliers for (1.1), with[1] $I(X; \hat{X}) + \beta \mathbb{E} [d(x, \hat{x})]$ and $\beta > 0$, one obtains a pair of equations for $p(\hat{x}|x)$ and the marginal $p(\hat{x})$. Iterating over these equations boils down to the Blahut-Arimoto algorithm, [Cover and Thomas, 2006, 10.7-10.8]. That is, a necessary condition for a distribution to achieve the minimum at (1.1) is that it is a fixed point of BA, or equivalently a root of the operator

$$F := Id - BA_{\beta} , \tag{1.2}$$

as noted by Agmon et al.. Where, BA_{β} denotes a single BA iteration at the multiplier value β, and Id is the identity. The marginal $p(\hat{x})$ may be taken as our variable (see Section I.2.3). However, to facilitate the discussion we shall write $\boldsymbol{x} \in \mathbb{R}^T$ instead, for some $T > 0$. Namely, we consider roots of an equation

$$F(\boldsymbol{x}, \beta) = \boldsymbol{0} , \tag{1.3}$$

---

[1]The normalization constraint is omitted for clarity.

for an operator $F(\cdot, \beta)$ on $\mathbb{R}^T$, $F : \mathbb{R}^T \times \mathbb{R} \to \mathbb{R}^T$, and β a real independent "time-like" parameter.

This work stems from the following intuition. Suppose that the Jacobian matrix $D_{\boldsymbol{x}} F$ of F is non-singular at a root $(\boldsymbol{x}_0, \beta_0)$ of (1.3). Then, by the Implicit Function Theorem, there not only exists a function $\boldsymbol{x}(\beta)$ satisfying (1.3) through the root, $\boldsymbol{x}(\beta_0) = \boldsymbol{x}_0$, but $\boldsymbol{x}(\beta)$ also inherits the differentiability properties of F, [Kielhöfer, 2012, I.1.7]. For example, $\boldsymbol{x}$ is analytic in β if F is analytic. When F is particularly well-behaved, then one might expect the derivatives of $\boldsymbol{x}$ with respect to β determine the path $\boldsymbol{x}(\beta)$. Indeed, when F is real-analytic as in rate-distortion (1.2), then each coordinate of $\boldsymbol{x}(\beta)$ can be written as power-series around β_0. If the series' convergence radii happen to be large or infinite, then the entire solution path $\boldsymbol{x}(\beta)$ can be extrapolated from the point $(\boldsymbol{x}_0, \beta_0)$, at least in principle. But even if these radii are small (but non-zero), determining $\boldsymbol{x}(\beta)$ only in some small vicinity of β_0, then one can extrapolate the path $\boldsymbol{x}(\beta)$ segment by segment, so long that $D_{\boldsymbol{x}} F$ remains non-singular.

We provide three novelties to transform this intuition into an arguably practical algorithm. The derivatives of $\boldsymbol{x}$ with respect to β are *implied* by the requirement $F = \boldsymbol{0}$ (1.3). To calculate implicit derivatives, we first provide a recursive formula (Theorem 3 in Section I.2.2) for the implicit derivatives $\frac{d^l}{d\beta^l} \boldsymbol{x}$ at a root in terms of the derivative tensors of F, for any order $l > 0$. While Zemel [2019] already solved this for a univariate $\boldsymbol{x}$, we are unaware of such formulae when $\boldsymbol{x}$ is multivariate. Second, we provide arbitrary-order closed-form formulae (Theorem 4 in Section I.2.3) for the derivative tensors of rate-distortion (1.2). Agmon et al. [2021] already provided its Jacobian, while for the related channel capacity Yu [2010] provided the Jacobian and Nakagawa et al. [2021] also the Hessian.

Together, the first two components allow us to calculate numerically the implicit derivatives

$$\left. \frac{d^l \boldsymbol{x}}{d\beta^l} \right|_{(\boldsymbol{x}_0, \beta_0)} \tag{1.4}$$

of arbitrary order at a root of rate-distortion (1.2), under mild assumptions. In fact, we have discovered a first-order ordinary differential equation (ODE) satisfied by RD roots (Theorem 14 in Section II.5.3). Using the implicit derivatives (1.4) we can approximate $\boldsymbol{x}(\beta)$ nearby, for example via

$$\boldsymbol{x}(\beta_0 + \Delta\beta) \cong \frac{1}{0!} \boldsymbol{x}_0 + \frac{1}{1!} \left. \frac{d\boldsymbol{x}}{d\beta} \right|_{(\boldsymbol{x}_0, \beta_0)} \Delta\beta + \frac{1}{2!} \left. \frac{d^2 \boldsymbol{x}}{d\beta^2} \right|_{(\boldsymbol{x}_0, \beta_0)} \Delta\beta^2 + \cdots + \frac{1}{l!} \left. \frac{d^l \boldsymbol{x}}{d\beta^l} \right|_{(\boldsymbol{x}_0, \beta_0)} \Delta\beta^l , \tag{1.5}$$

where $\Delta\beta := \beta - \beta_0$. One can then take a step $\Delta\beta$ and recompute expansion (1.5) repeatedly. This simple algorithm based on the Taylor method [Butcher, 2016, Atkinson et al., 2011] gives a piecewise polynomial approximation of the path $\boldsymbol{x}(\beta)$. After computing (1.5) along a grid, then any off-grid point is obtained by merely evaluating a polynomial. e.g., the right of Figure 1.1. With that, the above is not to be confused with the gradient flow towards a root at a fixed multiplier value β_0, which [Parker et al., 2010] describe in a related context by an ODE. In contrast, the implicit derivatives (1.4) describe how a root evolves *with* β.

While the above goes a long way towards reconstructing the entire solution path, it is not enough. For, as we show in Section II.6, RD problems typically have a plethora of sub-optimal roots, *not* achieving the minimum in (1.1). RD roots can collide and merge into one, or exchange optimality in some cases where the RD curve has a linear segment. These are instances of *bifurcations* — a change in the number of roots (1.3). To ensure that the root being traced is optimal, some understanding of the solution structure is necessary. That is, of the bifurcations of the fixed-point equations of RD, encoded by (1.2). Together with the two components above, the understanding established in Section II.6 provides tools to detect and handle bifurcations (of some types), culminating in our algorithm for tracking the path of an optimal RD root (Algorithm 3 in Section I.3.2), subject to mild assumptions. In addition to the interest in rate distortion itself, this
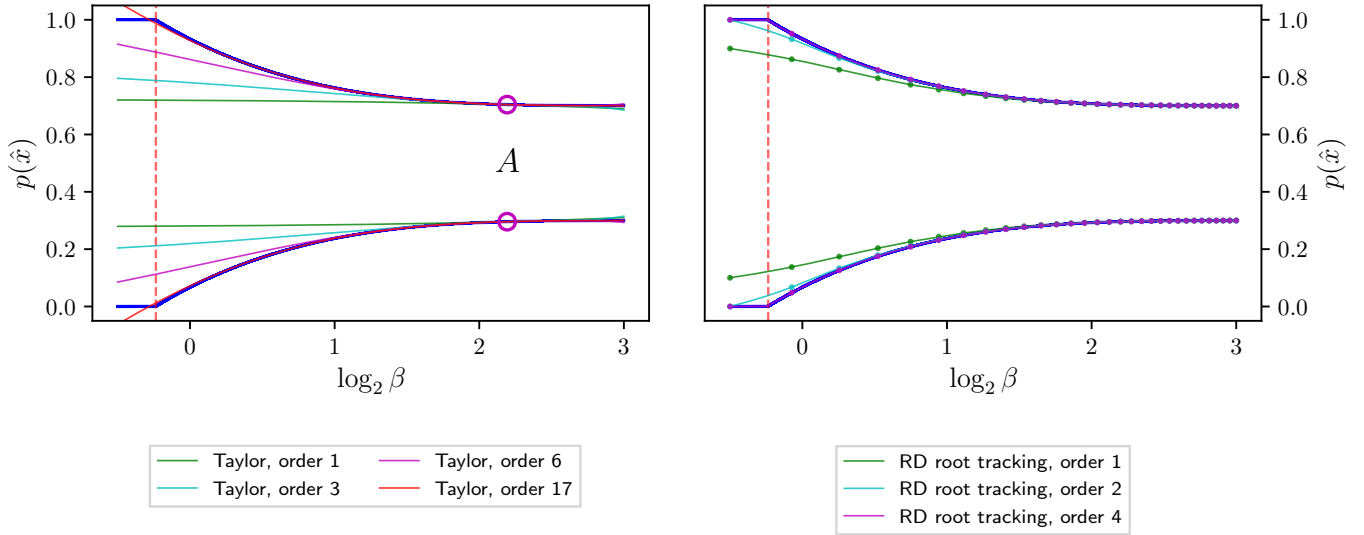
Figure 1.1: **Approximating the entire solution curve with implicit derivatives** (1.4), for a binary source with a Hamming distortion, compared to its analytical solution in Section III.F (thick blue). Each color depicts a marginal probability distribution $p(\hat{x})$ as a function of the Lagrange multiplier β, for $\hat{x} \in \{1, 2\}$; this suffices to parametrize a root (cf., Section I.2.3). **Left**: Taylor expansions (1.5) of several orders around the point A. **Right**: root-tracking for RD (Algorithm 3, with $\delta = 10^{-2}$) detects and handles the bifurcation (dashed red vertical). Adding grid-points bootstraps the approximation towards higher accuracy, with each grid point leveraging the computational cost invested earlier, to its right. Unlike BA, the entire solution curve is extrapolated here from only 31 grid points.

provides an example of how an understanding of a problem's bifurcations can be translated to a numerical algorithm. As the computation of the rate-distortion function is equivalent to a process of deterministic annealing [Rose, 1998], we believe that similar tools might facilitate the numerical solution of other problems as well.

RD bifurcations were noted already by Berger [1971] and others due to the resulting non-smoothness of the RD curve (1.1) at the points of bifurcation. Rose [1994] uses a mapping approach to provide insights for continuous source alphabets, usually assuming a squared-error distortion, $d(x, \hat{x}) = |x - \hat{x}|^2$. While allowing for a much broader class of distortion measures, our results in Section II.6 suggest a slightly different picture of RD bifurcations for sources of finite alphabet. Further, they allow a clearer view of cases where BA with reverse deterministic annealing [Rose et al., 1990, Rose, 1998] follows a sub-optimal solution path rather than the optimal one.

Our algorithm is perhaps best compared to solving an ODE numerically. Where, one usually exploits derivatives only to the order dictated by the ODE in order to propagate the solution. On the other hand, the mathematical machinery we provide allows us to compute implicit multivariate derivatives $\frac{d^l \boldsymbol{x}}{d\beta^l}$ (1.4) of arbitrary order, which we specialize to RD. In principle, one may even change the order at will with this machinery. With that, we have made several deliberate concessions for the sake of simplicity. Among others, fixing the order and step-size results in inefficient use of computational resources; see Section I.3.4 on various possible improvements and II.7 on the root cause of the computational difficulty. Nevertheless, despite these concessions, the cost of an entire solution curve appears to be roughly comparable to BA with reverse annealing, as suggested by Figure 1.2. Furthermore, we handle only cluster-vanishing bifurcations as in [Agmon et al., 2021], although our understanding of RD bifurcations (in Section II.6) permits more than that.

The error of an l-th order Taylor method for RD, as in expansion (1.5), is of order $O(|\Delta\beta|^l)$ for small step sizes $|\Delta\beta|$ (Theorem 5 in Section I.3.1); cf., Figure 1.1. Increasing l or taking $|\Delta\beta|$ smaller improves the approximation in general, as one might expect. Though, for simplicity, we fix these parameters while computing. This method can be seen to have better accuracy in general than interpolating the output distributions of BA, for example, at least for orders $l > 1$. Though, the details of this are deferred to future work. The computational costs of Algorithm 3 are only linear in source alphabet size $|\mathcal{X}|$ thanks to our choice of coordinates for $\boldsymbol{x}$ (in Section I.2.3), and are asymptotically polynomial in $|\hat{\mathcal{X}}|$ when l is fixed. On the other hand, its computational costs grow (hyper-)exponentially with the order l. However, despite this rapid growth rate, higher orders l make much better use of the invested computational costs, in general, when the step size $|\Delta\beta|$ is small. The cost-to-error tradeoff is demonstrated in Figure 1.2, with details in Section I.3.3.

To assist the reader, this paper is divided into three. Part I focuses on the necessary details, such as how to compute implicit derivatives for RD problems, and how these could be used to reconstruct the solution curve. Supporting ideas are elaborated in Part II: an outline of the derivation of the formulae for RD derivative tensors, a study of RD bifurcations and their relations to root-tracking, and complexity and error analyses. Part III provides the proofs and technical details omitted elsewhere. At the beginning of most parts and sections, we placed a short overview to help keep track of the logical flow. We also provide an annotated source code of our implementation at https://github.com/shlomiag/RTRD

Notations. Vectors are denoted boldface $\boldsymbol{x}$, its j-th coordinate x_j and scalars x in regular-font. An initial vector condition is denoted $\boldsymbol{x}_0$. Also, $\boldsymbol{\alpha} = (\alpha_0, \boldsymbol{\alpha}_+)$ when considering α_0 as the zeroth coordinate of $\boldsymbol{\alpha}$, and $\boldsymbol{\alpha}_+$ for the rest. $\Delta[S]$ denotes the *probability simplex* or *simplex* on a (finite) set S, $\{p \in \mathbb{R}^S : p(s) \geq 0 \text{ for } s \in S, \text{ and } \sum_{s \in S} p(s) = 1\}$. T is the dimension of an unspecified operator $F(\boldsymbol{x}, \beta)$, $\boldsymbol{x} \in \mathbb{R}^T$, as in (1.3). $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$ are the source and reproduction alphabet sizes of a given RD problem (d, p_X). A reproduction symbol $\hat{x} \in \hat{\mathcal{X}}$ is also called a *cluster*.
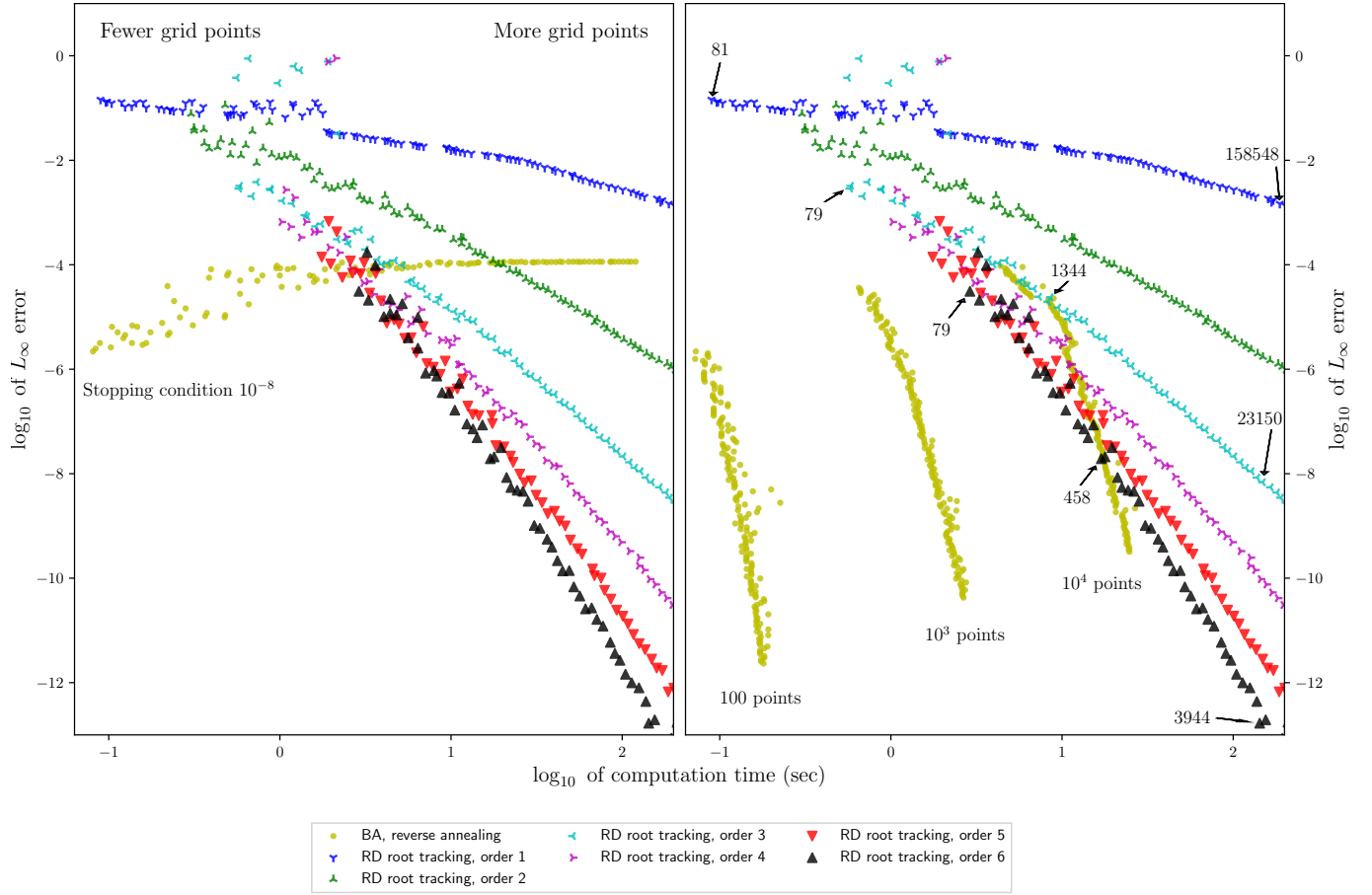
Figure 1.2: **The tradeoff between error and computational cost, by the number of grid points**. For each number of equally spaced grid points, the maximal error from the true solution (over all grid points) is plotted against the computation time, which serves as a rough measure for the computational complexity of the entire solution curve. Computation time was measured with our implementation, running single-threaded on a 1.80GHz Intel i7-8550U processor; see I.3.3 on complexities and cost-to-error tradeoff. Results are shown for RD root tracking (Algorithm 3) of several orders, and for Blahut-Arimoto computed in reverse annealing. Unlike BA, RD root tracking approximates the whole solution curve rather than just the grid points. The leftmost marker for each algorithm represents a grid of about 80 points, with an ∼ 8% increase between consecutive markers. When there are too few grid points, RD root tracking is sensitive to their precise location, often failing to detect the bifurcation. This is manifested by large errors to the top-left. Error calculation for root tracking ignores the point of heuristic itself, where a cluster mass threshold of $\delta = 0.01$ is crossed (see Section I.3.2). Solutions along the grid are compared against the analytical ones for a binary source with a Hamming distortion, Section III.F. Grid points were selected uniformly for all algorithms in an attempt to avoid bias in error estimation. **Left**: BA's grid size is increased gradually, from only 80 grid-points at the left to ∼ 760,000 at the right, with a stopping condition of 10^{-8}. As the grid becomes denser, BA computes at points closer to the bifurcation. Due to BA's critical slowing down, its accuracy is therefore reduced as the grid becomes denser, which is clearly noticed to the right. cf., the bottom of Figure 2.3. **Right**: BA's stopping condition is varied gradually from 10^{-8} to ∼ 10^{-14}, for a grid of fixed sizes $100, 10^3$ or 10^4. The number of grid points is shown at several markers, for RD root tracking of orders 1, 3, and 6. Notice the intersection of the plot for BA with 10^4 grid-points with RD root tracking of order 6; the latter obtains the entire solution curve with only 458 points, at the same cost and accuracy.

Part I
How to track operator roots for rate-distortion problems

This part aims to present the details necessary for our root-tracking Algorithm 3, which reconstructs the solution curve of a rate-distortion problem.

To that end, we start in 2.1 with the mathematical observation underlying the calculation of higher implicit multivariate derivatives at an operator root. This is accompanied by direct calculations of low-orders, and a simple yet non-trivial example in 2.1.1, before diving in 2.2 into the machinery for implicit derivatives of arbitrary order. In 2.3, we provide closed-form formulae for the derivative tensors of the Blahut-Arimoto operator. This allows us to specialize the above machinery to implicit derivatives of rate-distortion problems of arbitrary order.

Once the results for implicit derivatives at a point are in place, we use them in 3.1 to reconstruct the entire solution curve. This requires an understanding of the bifurcations of the operator at hand. For rate-distortion problems, we build on the results of Agmon et al. [2021], which are expanded significantly in Section II.6. This allows RD bifurcations to be handled in 3.2, completing the algorithm. Error guarantees for our algorithm, its computational and memory complexities, and its error-complexity tradeoff are discussed in 3.3. We conclude this part by discussing in 3.4 how our choices could be improved to yield a more efficient variant of the proposed algorithm.

Part II expands on these results, explaining details that are important yet not strictly necessary for understanding the algorithm.

2 Implicit derivatives at an operator's root, and for rate-distortion problems

This section provides the tools to calculate implicit derivatives at a root $(\boldsymbol{x}_0, \beta_0)$ of an arbitrary operator F, culminating with the results necessary for implicit derivatives of rate-distortion problems.

2.1 How implicit derivatives at an operator's root can be calculated

The sequel rests upon the following observation. Let $F(\cdot, \beta)$ be an operator on $\mathbb{R}^T$, $F : \mathbb{R}^T \times \mathbb{R} \to \mathbb{R}^T$, and let $(\boldsymbol{x}_0, \beta_0)$ be its root, $F(\boldsymbol{x}_0, \beta_0) = \boldsymbol{0}$ (1.3), such that

Assumption 1 ($\boldsymbol{x}$ can be written as a function of β)**.** *There exists a function $\boldsymbol{x}(\beta)$ defined in some neighborhood of β_0, such that $\boldsymbol{x}(\beta)$ is a root of F through $\boldsymbol{x}_0$.*

Assumption 2. *The function $\boldsymbol{x}(\beta)$ is differentiable at β_0 as many times as needed.*

We shall assume throughout that any derivative of F exists, as in the case of the Blahut-Arimoto operators. Using $\boldsymbol{x}(\beta)$ from Assumption 1, we have a well-defined composition

$$\beta \longmapsto \big(\boldsymbol{x}(\beta), \beta\big) \longmapsto F\big(\boldsymbol{x}(\beta), \beta\big) \tag{2.1}$$

in the vicinity of β_0. Since $\big(\boldsymbol{x}(\beta), \beta\big)$ is a root, this composition is nothing but the constant path $\beta \mapsto \boldsymbol{0}$ in $\mathbb{R}^T$. Hence, by Assumption 2, its derivatives with respect to β must vanish, to any order. That is,

$$\frac{d^l}{d\beta^l} F\big(\boldsymbol{x}(\beta), \beta\big)\Big|_{\beta_0} = \boldsymbol{0}, \tag{2.2}$$

for any $l \geq 0$. In essence, all the results below are encoded in Equation (2.2), with the remainder of this section dedicated to extracting the information of interest.

Before specifying *how* implicit derivatives $\frac{d^l \boldsymbol{x}}{d\beta^l}$ (1.4) can be calculated from (2.2), several clarifications are due. We shall require throughout that Assumptions 1 and 2 hold at any root of F, except perhaps at points of bifurcation. However, we do *not* require a root $\boldsymbol{x}(\beta)$ at 1 to be unique. That is, there may be multiple functions $\boldsymbol{x}(\beta)$, all of which are roots of F (1.3) through $(\boldsymbol{x}_0, \beta_0)$. A-priori, it is not clear that $\boldsymbol{x}$ can be written as a function of β, nor that $\boldsymbol{x}(\beta)$ is sufficiently differentiable. Both of these assumptions follow from the Implicit Function Theorem; e.g., [Kielhöfer, 2012, Theorem I.1.1]. For, write $D_{\boldsymbol{x}}F(\boldsymbol{x}_0, \beta_0)$ for the Jacobian matrix of F with respect to its $\boldsymbol{x}$ coordinates, evaluated at $(\boldsymbol{x}_0, \beta_0)$. Recall, it is the matrix defined by $(D_{\boldsymbol{x}}F)_{i,j} := \frac{\partial F_i}{\partial x_j}$. When it is non-singular, then the theorem not only implies that there exists a unique function $\boldsymbol{x}(\beta)$ through $(\boldsymbol{x}_0, \beta_0)$, but also that $\boldsymbol{x}(\beta)$ inherits the differentiability properties of F, [Kielhöfer, 2012, I.1.7]. That is, the derivatives $\frac{d^l \boldsymbol{x}}{d\beta^l}\big|_{\beta_0}$ (1.4) exist, to any order. However, we shall usually only require that Assumptions 1 and 2 hold, rather than than the stronger condition that $D_{\boldsymbol{x}}F$ is non-singular. This is reasonable for rate-distortion problems — see Section 3.1 for details. In general, however, understanding the bifurcations of F is necessary — e.g., Section II.6 for RD. For example, both assumptions hold for the constant one-dimensional operator $F(x, \beta) := 0$, whose Jacobian vanishes everywhere despite having no bifurcations. While on the other hand, the Jacobian of Example 2.1.1 below is singular precisely at the point of bifurcation there, where its two solution curves intersect and annihilate each other (see Figure 2.1).

We start with a gentle low-dimensional introduction as to how the derivatives $\frac{d^l \boldsymbol{x}}{d\beta^l}$ (1.4) can be calculated. Write $F = (F_1, \ldots, F_T)$ for the coordinates of F. Applying the multivariate chain-rule to the first-order equation $\frac{d}{d\beta}F = \boldsymbol{0}$ (2.2) reads,

$$\frac{d}{d\beta}F_i = \sum_{j=1}^{T} \frac{\partial F_i}{\partial x_j}\frac{dx_j}{d\beta} + \frac{\partial F_i}{\partial \beta} = 0 \tag{2.3}$$

for each $i = 1, \ldots, T$. Functions are understood henceforth to be evaluated at $(\boldsymbol{x}_0, \beta_0)$, unless otherwise stated. Write $D_\beta F$ for the vector whose i-th entry is $\frac{\partial F_i}{\partial \beta}$. The calculation of $\frac{d\boldsymbol{x}}{d\beta}$ thus boils down to solving the linear equation[2]

$$D_{\boldsymbol{x}}F \, \tfrac{d\boldsymbol{x}}{d\beta} = -D_\beta F \ . \tag{2.4}$$

This is an implicit ordinary differential equation (ODE), describing how the root $\boldsymbol{x}$ evolves with β. cf., Section 3.1 on the analyticity of RD roots. This ODE is an immediate consequence of the Implicit Function Theorem, [de Oliveira, 2014, Theorem 5], at least when the Jacobian is non-singular. We derive its explicit form (2.5) for RD later, in Theorem 14 (Section II.5.3). A marginal distribution $\boldsymbol{r}$ of full support on the reproduction alphabet $\hat{\mathcal{X}}$ which is a fixed-point of Blahut-Arimoto satisfies

$$\sum_{\hat{x}'} A_{\hat{x},\hat{x}'}\frac{dr(\hat{x}')}{d\beta} = \mathbb{E}_{p(\hat{x}',x)}\left[q(\hat{x}|x)d(x,\hat{x}')\right] - \mathbb{E}_{p_X}\left[q(\hat{x}|x)d(x,\hat{x})\right] \tag{2.5}$$

for every $\hat{x}$, where $A_{\hat{x},\hat{x}'}$ is given by (5.14), and $\boldsymbol{q}$ is the test channel or *encoder* corresponding to $\boldsymbol{r}$ by the BA Equation (2.18).

Before proceeding to the higher-order counterparts of the implicit ODE (2.4), some classic material on higher derivatives is necessary. e.g., [Dieudonné, 1969, VIII.12] or the very readable [Aguilar, 2021, 10.3] for the following. Let $f(\boldsymbol{x}, \beta)$ be a real-valued function on $\mathbb{R}^T \times \mathbb{R}$, $f : \mathbb{R}^T \times \mathbb{R} \to$

---

[2]It suffices to find a linear pre-image under $D_{\boldsymbol{x}}F$ if it is not invertible.

$\mathbb{R}$. Fixing β at β_0 for a moment, its gradient is the vector $(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_T})$. The gradient can be considered as a linear functional, mapping a vector $\boldsymbol{v} \in \mathbb{R}^T$ to $v_1 \cdot \frac{\partial f}{\partial x_1} + \cdots + v_T \cdot \frac{\partial f}{\partial x_T}$. This is useful when considering the first-order Taylor expansion $f(\boldsymbol{x}_0, \beta_0) + \sum_j v_j \frac{\partial f}{\partial x_j}$ of $f(\cdot, \beta_0)$ about $\boldsymbol{x}_0$, where $\boldsymbol{v} := \boldsymbol{x} - \boldsymbol{x}_0$ is the deviation from the point of expansion. Similarly, the Hessian matrix $\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)_{i,j=1}^T$ of f is a bi-linear map. Namely, it maps a pair of vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^T$ to $\sum_{i,j} u_i v_j \cdot \frac{\partial^2 f}{\partial x_i \partial x_j}$, in a manner which is separately linear in each of the two vectors. The second-order expansion of f maps a deviation $\boldsymbol{v} \in \mathbb{R}^T$ from the basepoint $\boldsymbol{x}_0$ to $f(\boldsymbol{x}_0, \beta_0) + \sum_j v_j \frac{\partial f}{\partial x_j} + \sum_{i,j} v_i v_j \cdot \frac{\partial^2 f}{\partial x_i \partial x_j}$.

Next, replace the real-valued f with the i-th component F_i of the operator $F = (F_1, \ldots, F_T)$ that we have started with, and allow derivatives also with respect to β. The above generalizes as follows. For orders $b, m \geq 0$ and a coordinate $1 \leq i \leq T$ (all integers), denote by $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i$ the following symmetric[3] m-multilinear map, which is defined on $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m \in \mathbb{R}^T$ by

$$D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i \Big|_{(\boldsymbol{x}_0, \beta_0)} [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] := \sum_{1 \leq i_1, i_2, \ldots, i_m \leq T} v_{1,i_1} \cdots v_{m,i_m} \cdot \frac{\partial^m}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_m}} \frac{\partial^b F_i}{\partial \beta^b} (\boldsymbol{x}_0, \beta_0) \ .$$

(2.6)

Just as in the second-order Hessian, each index i_j varies independently over all the coordinates $1, \ldots, T$ of $\boldsymbol{v}_j$, for $j = 1, \ldots, m$. We write $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F$ for the vector whose i-th coordinate is $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i$. Evaluation at $(\boldsymbol{x}_0, \beta_0)$ shall be omitted whenever clear from the context. For practical purposes, the i-th multilinear map $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i$ may be considered as a "matrix" with m axes, and $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F$ as a "matrix" with $m+1$ axes. For example, the vector $D_{\boldsymbol{x}} F_i$ is the gradient of the (scalar) function F_i, while $D_{\boldsymbol{x}} F$ is the T-by-T Jacobian matrix of F. The integers b and m are the orders of differentiation with respect to β and the coordinates of $\boldsymbol{x}$, respectively. For $b > 0$, $D^{b+1}_{\beta^b, \boldsymbol{x}} F_i$ and $D^{b+1}_{\beta^b, \boldsymbol{x}} F$ are derivative tensors of higher order, although of the same respective sizes as $D_{\boldsymbol{x}} F_i$ and $D_{\boldsymbol{x}} F$. When b and i are given, one can think of the m-tuples $(i_1, i_2, \ldots, i_m)$ indexing $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i$ in (2.6) as differentiations in some particular order. First derive $\frac{\partial^b F_i}{\partial \beta^b}$ with respect to x_{i_m}, then with respect to $x_{i_{m-1}}$, and so forth up to x_{i_1}. This notation is redundant, in the sense that exchanging distinct indices $i_{j_1} \neq i_{j_2}$ yields the same partial derivative when F_i is well-behaved.

When i is understood from the context, it is sometimes convenient to index $D^{b+m}_{\beta^b, \boldsymbol{x}^m} F_i$ by a multi-index $\boldsymbol{\alpha} = (\alpha_0, \boldsymbol{\alpha}_+) \in \mathbb{N}_0 \times \mathbb{N}_0^T$, with $\mathbb{N}_0$ the non-negative integers. α_0 then represents the number b of differentiations with respect to β, and α_j the number of differentiations with respect to x_j for $1 \leq j \leq T$. Define $|\boldsymbol{\alpha}| := \sum_j \alpha_j$. When setting $\boldsymbol{\alpha} = (1, 3, 2) \in \mathbb{N}_0^{1+2}$ for an operator F on $T = 2$ dimensions, $\frac{\partial^6}{\partial \beta \, \partial x_1^3 \partial x_2^2} F_i$ for example is shortened to $\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \beta \partial \alpha_0 \partial \boldsymbol{x}^{\boldsymbol{\alpha}_+}} F_i = \frac{\partial^6}{\partial \beta \partial \boldsymbol{x}^{(3,2)}} F_i$, for $i = 1, 2$. Because the order of differentiation does not matter when F_i is well-behaved, then $\boldsymbol{\alpha}$ corresponds not only to the entry $(i_1, \ldots, i_5) = (1, 1, 1, 2, 2)$ of $D^6_{\beta^1, \boldsymbol{x}^5} F_i$ (2.6), but also to its permutations. See also the multivariate notation definitions (4.3), in Section II.4.

At the next order, differentiating $\frac{d}{d\beta} F_i = 0$ (2.3) with respect to β yields

$$\frac{d^2}{d\beta^2} F_i = \sum_{j,k} \frac{\partial^2 F_i}{\partial x_k \partial x_j} \frac{dx_j}{d\beta} \frac{dx_k}{d\beta} + \sum_j \frac{\partial^2 F_i}{\partial \beta \partial x_j} \frac{dx_j}{d\beta} + \sum_j \frac{\partial F_i}{\partial x_j} \frac{d^2 x_j}{d\beta^2} + \sum_j \frac{\partial^2 F_i}{\partial x_j \partial \beta} \frac{dx_j}{d\beta} + \frac{\partial^2 F_i}{\partial \beta^2} = 0 \ . \quad (2.7)$$

Unlike the first-order expansion (2.3), the second-order one (2.7) contains a 3-tensor term $\sum \frac{\partial^2 F_i}{\partial x_k \partial x_j} \frac{dx_j}{d\beta} \frac{dx_k}{d\beta}$. By definition (2.6), the latter is $D^2_{\boldsymbol{x}, \boldsymbol{x}} F_i[\frac{d\boldsymbol{x}}{d\beta}, \frac{d\boldsymbol{x}}{d\beta}]$, while the mixed-derivatives term following it is

---

[3] An m-multilinear map T is *symmetric* if its value is unchanged by an arbitrary permutation of its arguments: $T[\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] = T[\boldsymbol{v}_{\sigma(1)}, \ldots, \boldsymbol{v}_{\sigma(m)}]$ for any permutation σ of $\{1, 2, \ldots, m\}$.

$D^2_{\beta,\boldsymbol{x}}F_i[\frac{d\boldsymbol{x}}{d\beta}]$, and so forth. Rewriting the first few expansion orders $\frac{d^l F}{d\beta^l} = \boldsymbol{0}$ (2.2) in this notation,

$$\boldsymbol{0} = \tfrac{d^0 F}{d\beta^0} = F \tag{1.3}$$

$$\boldsymbol{0} = \tfrac{d^1 F}{d\beta^1} = D_{\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}] + D_{\beta}F \tag{2.8}$$

$$\boldsymbol{0} = \tfrac{d^2 F}{d\beta^2} = D_{\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}] + D^2_{\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}, \tfrac{d\boldsymbol{x}}{d\beta}] + 2D^2_{\beta,\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}] + D^2_{\beta,\beta}F \tag{2.9}$$

$$\boldsymbol{0} = \tfrac{d^3 F}{d\beta^3} = D_{\boldsymbol{x}}F[\tfrac{d^3\boldsymbol{x}}{d\beta^3}] + D^3_{\boldsymbol{x},\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}, \tfrac{d\boldsymbol{x}}{d\beta}, \tfrac{d\boldsymbol{x}}{d\beta}] + 3D^2_{\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}, \tfrac{d\boldsymbol{x}}{d\beta}]$$
$$+ 3D^3_{\beta,\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}, \tfrac{d\boldsymbol{x}}{d\beta}] + 3D^2_{\beta,\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}] + 3D^3_{\beta,\beta,\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}] + D^3_{\beta,\beta,\beta}F \tag{2.10}$$

where the third-order expansion (2.10) for $\frac{d^3}{d\beta^3}F$ follows from (2.9) by a straightforward calculation.

As can be seen in (2.8)-(2.10), the implicit derivative $\frac{d^l\boldsymbol{x}}{d\beta^l}$ (1.4) of highest-order appears only once in the l-th order expansion, at the product $D_{\boldsymbol{x}}F[\frac{d^l\boldsymbol{x}}{d\beta^l}]$ of $\frac{d^l\boldsymbol{x}}{d\beta^l}$ by the linear map $D_{\boldsymbol{x}}F$. While, the other terms in each equation contain only derivatives $\frac{d^k\boldsymbol{x}}{d\beta^k}$ of lower orders $0 < k < l$. This holds for any order $l > 0$, as we show in Section 2.2 below. So, in principle, all one needs to do in order to obtain the derivatives $\frac{d^l\boldsymbol{x}}{d\beta^l}$ (1.4) of *any* order $l > 0$ is to solve these equations recursively:

1. Suppose that the lower-order derivatives $\frac{d^k\boldsymbol{x}}{d\beta^k}$ are known for $0 < k < l$.

2. Calculate all the derivative tensors of F, up to order l.

3. Evaluate the multilinear forms in the expansion of $\frac{d^l}{d\beta^l}F = \boldsymbol{0}$, except for $D_{\boldsymbol{x}}F[\frac{d^l\boldsymbol{x}}{d\beta^l}]$.

4. Solve a linear equation in $\frac{d^l\boldsymbol{x}}{d\beta^l}$ with coefficients $D_{\boldsymbol{x}}F$.

Up to the technicalities of a general-order machinery this is the heart of Algorithm 1 below, for computing implicit multivariate derivatives of arbitrary order. Once these technicalities are settled in Section 2.2, there are two main tasks that one needs to tackle in order to obtain the derivatives $\frac{d^l\boldsymbol{x}}{d\beta^l}$ (1.4) at an operator root, and to be able to use them. First, one needs to calculate the derivative tensors of F, as in point 2 above. We have accomplished this for rate-distortion problems, providing in Section 2.3 closed-form formulae for the derivative tensors of $Id - BA_\beta$ (1.2), of any order. Second, one needs to tell whether the Taylor series (1.5) for $\boldsymbol{x}(\beta)$ around β_0 indeed approximates the true solution. For, operator roots may cease to be a function of β, cease to be differentiable, or even cease to exist at critical β values. This boils down to understanding the solution structure of F, or equivalently its bifurcations. While 2.1.1 below provides an example where Assumption 1 breaks (two roots collide and annihilate), the discussion for rate-distortion is subtler. We provide guarantees in Section 2.3, with the full details postponed till Section II.6.

2.1.1 Example: implicit derivatives of line intersections with a parabola

Before presenting in Section 2.2 the fully fledged machinery of arbitrary order, we give a simple yet non-trivial example how implicit derivatives $\frac{d^l\boldsymbol{x}}{d\beta^l}$ can be calculated at an operator root.

A line $y = ax + \beta$ in the plane typically intersects a parabola $y = bx^2 + cx + d$ (with $b \neq 0$) at either two points or none, as in Figure 2.1 for instance. In the special case when it is tangent to the parabola, it intersects at a single point. Consider the problem of tracking the intersection point as the line is being translated by varying β.

Setting

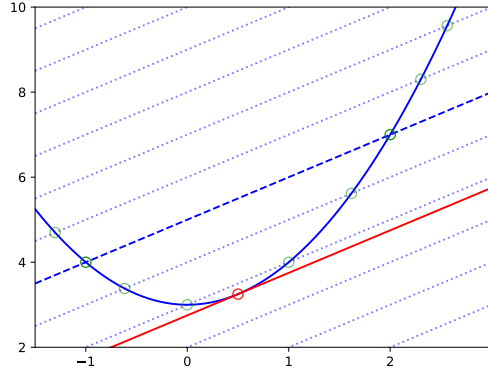$$F(x, y; \beta) := \begin{pmatrix} -y + bx^2 + cx + d \\ -y + ax + \beta \end{pmatrix}, \tag{2.11}$$

Figure 2.1: **Intersections of $y = x^2 + 3$ with the lines $y = x + \beta$.** The line at $\beta = 5$ is dashed blue, and at other integral β values in dotted blue. At $\beta_c = 2.75$ (solid red line) the system $F = \mathbf{0}$ (2.11) undergoes a bifurcation: there are two distinct roots above β_c, none below it, and exactly one at the bifurcation point itself.

the intersection point is encoded by requiring $F = \mathbf{0}$. Assume that (x_0, y_0) is known to be an intersection point at β_0, $F(x_0, y_0; \beta_0) = \mathbf{0}$, and that at the vicinity of (x_0, y_0), the intersection point can be written as a function of β, $(x, y) = \big(x(\beta), y(\beta)\big)$. By calculating the derivative tensors of F (2.11) and plugging them into the first few expansions of $\frac{d^l F}{d\beta^l} = \mathbf{0}$ (e.g., (2.8)-(2.10)), we can solve for the implicit derivatives $\frac{d^l x}{d\beta^l}, \frac{d^l y}{d\beta^l}$ at $(x_0, y_0; \beta_0)$. See Section III.A for full details. Doing so till fourth order yields a Taylor expansion (1.5) of the intersection point,

$$\begin{pmatrix} x(\beta) \\ y(\beta) \end{pmatrix} \approx \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \frac{1}{\Delta} \begin{pmatrix} 1 \\ a + \Delta \end{pmatrix} \cdot (\beta - \beta_0) - \frac{b}{\Delta^3} \begin{pmatrix} 1 \\ a \end{pmatrix} \cdot (\beta - \beta_0)^2$$
$$+ \frac{2b^2}{\Delta^5} \begin{pmatrix} 1 \\ a \end{pmatrix} \cdot (\beta - \beta_0)^3 - \frac{5b^3}{\Delta^7} \begin{pmatrix} 1 \\ a \end{pmatrix} \cdot (\beta - \beta_0)^4 \quad (2.12)$$

Where, $\Delta := 2bx_0 + c - a$ vanishes precisely when the slopes $2bx_0 + c$ of the parabola and a of the line coincide. At this point a bifurcation occurs, as can be seen in Figure 2.1: the two distinct intersection points of this problem merge then into one, and disappear beyond the critical β value. That is, the point of intersection (x_0, y_0) can no longer be written as a function of β. Figure 2.2 demonstrates the first few expansion orders (2.12), when expanded near or far of the bifurcation.

For this simple problem, the equation $F = \mathbf{0}$ (2.11) can also be solved analytically, yielding

$$x(\beta) = \frac{a-c}{2b} \pm \sqrt{\left(\frac{a-c}{2b}\right)^2 + \frac{\beta - d}{b}} \quad \text{and} \quad y(\beta) = \beta + ax(\beta) \quad (2.13)$$

Our approximation (2.12) is nothing but the fourth-order Taylor expansion of the exact solution (2.13) around β_0, as can be verified by expanding the latter directly. However, in contrast to the Taylor expansion of (2.13), the calculations leading to (2.12) can be carried out even when an exact solution is *not* available, as is typical for rate-distortion problems.

2.2 High-order implicit derivatives at an operator's root

We provide a machinery for calculating high-order implicit derivatives at an operator's root, built on the reasoning of Section 2.1. In particular, we assume henceforth that Assumptions 1 and 2
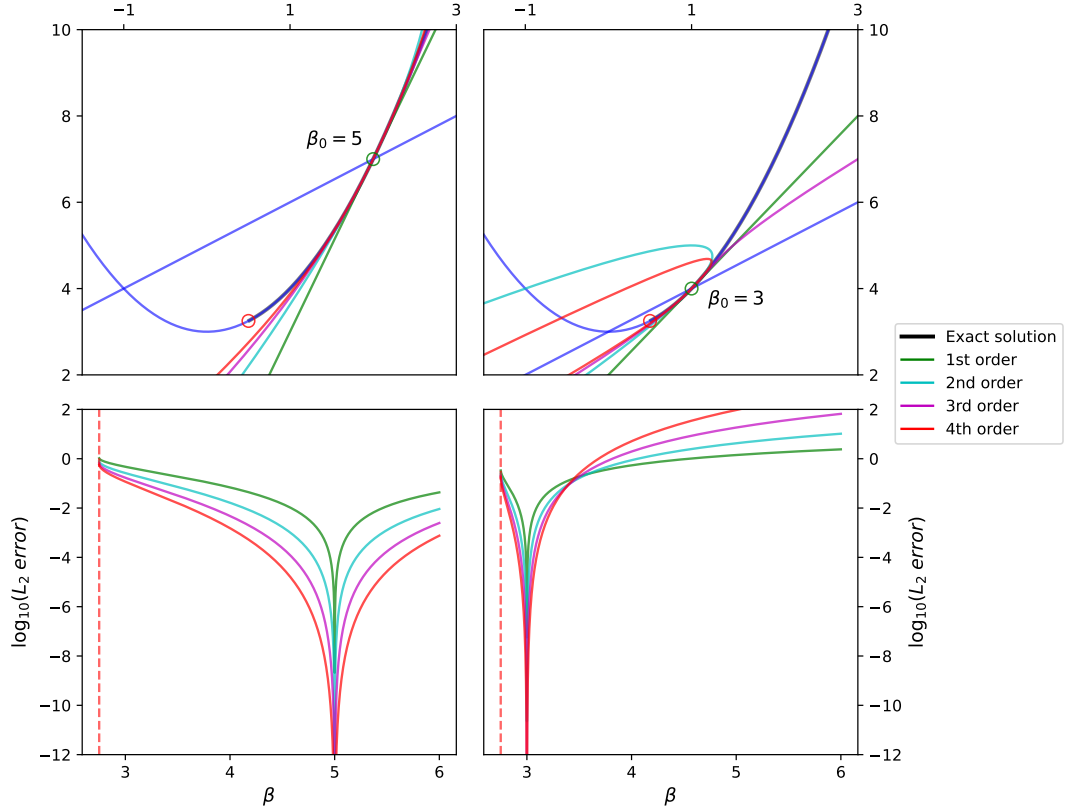
Figure 2.2: **Taylor approximations versus the exact solution** (2.13) **to** $F = 0$ (2.11), **by order and choice of base-point**, for the line-intersecting-parabola problem of Figure 2.1. **Top:** the expansions (2.12) of $(x(\beta), y(\beta))$ around β_0, colored by order, and the exact solution (2.13) in black. **Bottom:** the expansions' L_2-error from the exact solution. The comparison ends at the bifurcation point $\beta_c = 2.75$ (dashed red verticals), beyond which no solution exists. **Right:** the Jacobian $D_{\boldsymbol{x}} F$ becomes increasingly singular as the base-point β_0 is taken closer to β_c (see Section III.A). As a result, the implicit derivatives diverge and the approximations' quality deteriorates. cf., the comments after Theorem 3 (in Section 2.2) and Section II.7.3 on the latter implication.

there hold. As before, implicit derivatives and derivative tensors are understood to be evaluated at $(\boldsymbol{x}_0, \beta_0)$.

But first, a few definitions. A *partition* of an integer $n > 0$ is a sequence of positive integers $0 < p_1 \leq p_2 \leq \cdots \leq p_m$ whose sum is n, $\sum_{i=1}^{m} p_i = n$. The integers p_i are called the *parts* of the partition. The *partition function* $p(n)$ is the number of partitions of n. e.g., Andrews [1998]. When interested in the number of times m_i a part p_i appears (its *multiplicity*), we shall sometimes write $(m_1) \cdot p_1 + \cdots + (m_s) \cdot p_s$, with $0 < p_1 < \cdots < p_s$ now the partition's distinct parts. The *total multiplicity* is the number of parts $m := m_1 + \cdots + m_s$ in a partition. For example, there are three partitions of the integer $n = 3$, $p(3) = 3$. Namely, $1 + 2$, $1 + 1 + 1$, and the *trivial partition* 3. Written by multiplicity, these are $(1) \cdot 1 + (1) \cdot 2$, $(3) \cdot 1$ and $(1) \cdot 3$ respectively, of total multiplicities 2, 3 and 1.

We shall need to apply the m-multilinear maps $D_{\beta^b, \boldsymbol{x}^m}^{b+m} F_i$ (2.6) with repeated arguments. For a lack of better notation, if $\boldsymbol{v}_1$ appears as an argument m_1 times, $\boldsymbol{v}_2$ appears m_2 times, till $\boldsymbol{v}_s$

appearing m_s times, we denote

$$D^{b+m}_{\beta^b,\boldsymbol{x}^m}F_i[(\boldsymbol{v}_1)_{\times m_1},\ldots,(\boldsymbol{v}_s)_{\times m_s}] := D^{b+m}_{\beta^b,\boldsymbol{x}^m}F_i[\underbrace{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_1}_{m_1\text{ times}},\ldots,\underbrace{\boldsymbol{v}_s,\ldots,\boldsymbol{v}_s}_{m_s\text{ times}}] \tag{2.14}$$

With that, the arbitrary-order counterparts $\frac{d^l}{d\beta^l}F$ of (2.8)-(2.10) are as follows.

Theorem 1. *Let $F\big(\boldsymbol{x}(\beta),\beta\big)$ be the composition of $F:\mathbb{R}^T\times\mathbb{R}\to\mathbb{R}^T$ with a path $\boldsymbol{x}(\beta)$ in $\mathbb{R}^T$, both of which are sufficiently differentiable. Its derivative of order $l>0$ can be written as*

$$\frac{d^l}{d\beta^l}F(\boldsymbol{x}(\beta),\beta) = \sum_{partitions}\sum_{b=0}^{m_1\cdot\delta(p_1=1)}\frac{l!}{b!(m_1-b)!m_2!\cdots m_s!\cdot(p_1!)^{m_1}\cdots(p_s!)^{m_s}}$$
$$\cdot D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\frac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\frac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\ldots,\left(\frac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right] \tag{2.15}$$

where the outer sum is over the integer partitions $(m_1)\cdot p_1+\cdots+(m_s)\cdot p_s$ of l, $m := m_1+\cdots+m_s$, and δ is Dirac's delta function.

The inner summation at (2.15) goes up to the multiplicity of 1 in a partition, or zero if there is no part of size 1. This theorem is an application of a modern version of the multivariate Faà di Bruno's formula, Ma [2009], which is elaborated in Section II.4. See Section III.B.1 for a proof.

To illustrate Theorem 1, we order the summands in the expansion of $\frac{d^3}{d\beta^3}F$ by integer partitions. Recall the third-order expansion (2.10),

$$\frac{d^3 F}{d\beta^3} = D_{\boldsymbol{x}}F[\tfrac{d^3\boldsymbol{x}}{d\beta^3}] + D^3_{\boldsymbol{x},\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta}] + 3D^2_{\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2},\tfrac{d\boldsymbol{x}}{d\beta}]$$
$$+ 3D^3_{\beta,\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta}] + 3D^2_{\beta,\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}] + 3D^3_{\beta,\beta,\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}] + D^3_{\beta,\beta,\beta}F \ . \tag{2.16}$$

Each term above corresponds to a partition and to a number b of differentiations with respect to β, indexing the summations at (2.15). The part sizes and their multiplicities can be read off the tensors' arguments at (2.16). For example, $D_{\boldsymbol{x}}F[\tfrac{d^3\boldsymbol{x}}{d\beta^3}]$ corresponds to a partition with one part of size 3, and $b=0$ derivations with respect to β. While, $D^2_{\beta,\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}]$ corresponds to $(1)\cdot 1+(1)\cdot 2$, and $b=1$; a first-order argument $\tfrac{d\boldsymbol{x}}{d\beta}$ corresponding to the part of size 1 is left out because $b>0$. Proceeding this way, we obtain Table 1.

$l=3$	$(3)\cdot 1$	$(1)\cdot 1+(1)\cdot 2$	$(1)\cdot 3$	
	$D^3_{\boldsymbol{x},\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta}]$	$3D^2_{\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2},\tfrac{d\boldsymbol{x}}{d\beta}]$	$D_{\boldsymbol{x}}F[\tfrac{d^3\boldsymbol{x}}{d\beta^3}]$	$b=0$
	$3D^3_{\beta,\boldsymbol{x},\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta},\tfrac{d\boldsymbol{x}}{d\beta}]$	$3D^2_{\beta,\boldsymbol{x}}F[\tfrac{d^2\boldsymbol{x}}{d\beta^2}]$		$b=1$
	$3D^3_{\beta,\beta,\boldsymbol{x}}F[\tfrac{d\boldsymbol{x}}{d\beta}]$			$b=2$
	$D^3_{\beta,\beta,\beta}F$			$b=3$

Table 1: The summands of Theorem 1 for $l=3$. The expansion terms of $\frac{d^3 F}{d\beta^3}$ (2.16) are ordered by the integer partitions of 3 and by the number b of differentiations with respect to β.

No part in a partition of l can be of size larger than l. Clearly, the trivial partition $(1)\cdot l$ is the only one with a part of size l. Thus, among the derivatives $\frac{d^k\boldsymbol{x}}{d\beta^k}$ for $k>0$, $\frac{d^l\boldsymbol{x}}{d\beta^l}$ is that of highest-order which appears in $\frac{d^l}{d\beta^l}F$ (2.15). It appears there once, at the term indexed by the trivial partition $(1)\cdot l$ and $b=0$.

Corollary 2. *The derivative $\frac{d^l \boldsymbol{x}}{d\beta^l}$ appears once in the l-th order expansion (2.15) of $\frac{d^l}{d\beta^l}F$, at the term $D_{\boldsymbol{x}}F[\frac{d^l \boldsymbol{x}}{d\beta^l}]$. Any other term there contains only derivatives $\frac{d^k \boldsymbol{x}}{d\beta^k}$ of lower orders, $k < l$.*

With that, the derivative of highest-order $\frac{d^l \boldsymbol{x}}{d\beta^l}$ can be isolated easily from $\frac{d^l}{d\beta^l}F = \boldsymbol{0}$ (2.2). As an immediate consequence of Theorem 1, we obtain

Theorem 3. *Let $l > 1$, $\boldsymbol{x} = \boldsymbol{x}(\beta)$ a root of $F = \boldsymbol{0}$ (1.3), and assume that the derivatives $\frac{d^k \boldsymbol{x}}{d\beta^k}$ are known for all $0 < k < l$. Then,*

$$D_{\boldsymbol{x}}F\left[\frac{d^l \boldsymbol{x}}{d\beta^l}\right] = -\sum_{\substack{non\text{-}trivial \\ partitions}} \sum_{b=0}^{m_1 \cdot \delta(p_1=1)} \frac{l!}{b!(m_1-b)!m_2!\cdots m_s! \cdot (p_1!)^{m_1}\cdots(p_s!)^{m_s}}$$

$$\cdot D_{\beta^b, \boldsymbol{x}^{m-b}}^m F\left[\left(\frac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)}, \left(\frac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2}, \ldots, \left(\frac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right] \quad (2.17)$$

where the l-th order derivative $\frac{d^l \boldsymbol{x}}{d\beta^l}$ appears only at the left-hand side; the outer sum is over the non-trivial integer partitions of l. For $l = 1$, use $D_{\boldsymbol{x}}F[\frac{d\boldsymbol{x}}{d\beta}] = -D_{\beta}F$ (2.4) instead.

While Zemel [2019] provides a formula for implicit derivatives of arbitrary order when $\boldsymbol{x}$ is univariate, we are unaware of such a result in the literature for a multivariate $\boldsymbol{x}$. Unlike our Theorem 3 which is recursive and written as a sum over integer partitions, their formula (Theorem 15 therein) is explicit and written as a sum over vector partitions, while requiring that $D_{\boldsymbol{x}}F$ is non-singular. These differences make Theorem 3 computationally more efficient for our purposes, as we compute all the derivatives up to a given order, $\frac{d^k \boldsymbol{x}}{d\beta^k}$ for $k = 1, \ldots, l$. Our first-order expansion (2.8) coincides with that of Zemel [2019] when $\boldsymbol{x}$ is univariate; a direct exercise shows that so do our second and third order expansions (2.9)-(2.10) (see Equations (2)-(4) there). Verifying the equivalence directly for higher or is more challenging.

Now that we've established that $\frac{d^l}{d\beta^l}F = \boldsymbol{0}$ (2.2) can be solved for $\frac{d^l \boldsymbol{x}}{d\beta^l}$, we spell-out in Algorithm 1 the steps to solve it recursively from formula (2.17) in Theorem 3. The computational and memory complexities of Algorithm 1 are elaborated in Section II.8. It requires an auxiliary method (CALC DERIV TENSOR) to compute the derivative tensors (2.6) of F. For rate-distortion problems, these tensors are given in the next Section 2.3. When the latter are used, we refer to it as the specialization of Algorithm 1 to rate-distortion, or simply Algorithm 1 for RD. To keep things simple, Algorithm 1 is presented in a recursive form. Yet, implicit derivatives can be memorized (cached) at little memory cost $O(T \cdot l)$, thus avoiding the recursion. Similar comments go also to memorizing the derivative tensors of F — see Section 3.3 for details.

When F has multiple roots $\boldsymbol{x}(\beta)$ of through $(\boldsymbol{x}_0, \beta_0)$, then the Jacobian $D_{\boldsymbol{x}}F$ must be singular, by the Implicit Function Theorem. In accordance with that, formula (2.17) then has multiple solutions for each derivative order $\frac{d^l \boldsymbol{x}}{d\beta^l}$, as they are determined up to an element in $\ker D_{\boldsymbol{x}}F$. In the other way around, singularity of the Jacobian does *not* necessarily imply that there are multiple roots $\boldsymbol{x}(\beta)$ through $(\boldsymbol{x}_0, \beta_0)$. For, the existence of multiple solutions $\frac{d^l \boldsymbol{x}}{d\beta^l}$ to (2.17) need not imply that for each there is a root $\boldsymbol{x}(\beta)$ through $(\boldsymbol{x}_0, \beta_0)$ with the given implicit derivative. An understanding of the bifurcations of F is then necessary to determine which of the possible outputs of Algorithm 1 is indeed the implicit derivative of a root $\boldsymbol{x}(\beta)$ of $F = \boldsymbol{0}$ (1.3). We handle this in Algorithm 3 (Section 3.2) for tracking an RD root by ensuring that the Jacobian is non-singular. Algorithm 1 then has a unique and well-defined result, which by the Implicit Function Theorem must pertain to a root. See also Section II.6 on RD bifurcations.

The derivatives computed by Algorithm 1 lose their accuracy when a Jacobian eigenvalue approaches zero. e.g., when approaching a bifurcation. The effect is more pronounced as the

derivative's order increases. Figure 2.3 below demonstrates this nicely for rate-distortion. Its numerical error is negligible when far from bifurcation, and yet grows by the derivative's order when approaching it. This loss of accuracy can be traced back to the formula of Theorem 3. For, the Jacobian $D_{\boldsymbol{x}}F$ becomes ill-conditioned[4] when an eigenvalue gradually vanishes. So long that $D_{\boldsymbol{x}}F$ is invertible, the l-th order derivative $\frac{d^l \boldsymbol{x}}{d\beta^l}$ essentially contains $(D_{\boldsymbol{x}}F)^{-1}$ to the l-th power, amplifying numerical errors as l increases. This can be seen by multiplying both sides of (2.17) by $(D_{\boldsymbol{x}}F)^{-1}$, and then repeatedly substituting into $\frac{d^l \boldsymbol{x}}{d\beta^l}$ the explicit lower-order expressions for $\frac{d^k \boldsymbol{x}}{d\beta^k}$, for $k < l$.

Algorithm 1: High-order implicit derivatives of a root $\boldsymbol{x}(\beta)$ of an operator F

1: **function** CALCULATE IMPLICIT DERIVATIVE($\boldsymbol{x}_0, \beta_0, l; params$)

Input:
  Point of evaluation $(\boldsymbol{x}_0, \beta_0)$, derivative order $l > 0$,
  additional parameters $params$ needed for computing the derivative tensors.

Output: $\frac{d^l \boldsymbol{x}}{d\beta^l}\Big|_{(\boldsymbol{x}_0, \beta_0)}$, determined uniquely if $D_{\boldsymbol{x}}F$ is non-singular (see main text).

Require:
  A method CALC DERIV TENSOR($m, b; \boldsymbol{x_0}, \beta_0, params$) which computes $D^{m+b}_{\beta^b, \boldsymbol{x}^m} F\big|_{(\boldsymbol{x_0}, \beta_0)}$ (2.6).

2:   Initialize $result \leftarrow 0$.
3:   **for** $k = 1$ to $l - 1$ **do**
4:     $\frac{d^k \boldsymbol{x}}{d\beta^k} \leftarrow$ CALCULATE IMPLICIT DERIVATIVE($\boldsymbol{x}_0, \beta_0, k; params$)
      $\triangleright$ Cacheable; see main text.
5:   **end for**
6:   **for** partition $(m_1) \cdot p_1 + \cdots + (m_s) \cdot p_s$ of l **do**
7:     $b_{max} \leftarrow m_1$ if $p_1 = 1$ else 0.
8:     **for** $b = 0$ to b_{max} **do**
9:       $m \leftarrow m_1 + \cdots + m_s$
10:      **if** $m = 1$ and $b = 0$ **then** $\triangleright$ Corresponds to $D_{\boldsymbol{x}}F[\frac{d^l \boldsymbol{x}}{d\beta^l}]$, by Corollary 2.
11:        Continue
12:      **else**
13:        $coef \leftarrow \frac{l!}{b!(m_1-b)!m_2!\cdots m_s!\cdot(p_1!)^{m_1}\cdots(p_s!)^{m_s}}$
14:        $tensor \leftarrow$ CALC DERIV TENSOR($m - b, b; \boldsymbol{x}_0, \beta_0$) $\triangleright$ Cacheable; see main text.
15:        $result \leftarrow result - coef \cdot tensor\left[\left(\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)}, \left(\frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2}, \ldots, \left(\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]$
16:      **end if**
17:    **end for**
18:   **end for**
19:   $D_{\boldsymbol{x}}F \leftarrow$ CALC DERIV TENSOR($1, 0; \boldsymbol{x_0}, \beta_0$). $\triangleright$ Cacheable; see main text.
20:   $result \leftarrow$ a linear pre-image of $result$ under $D_{\boldsymbol{x}}F$.
21:   **return** $result$
22: **end function**

---

[4]That is, its condition number is large. Where the *condition number* is defined as the ratio between its largest and smallest eigenvalues, in absolute value.
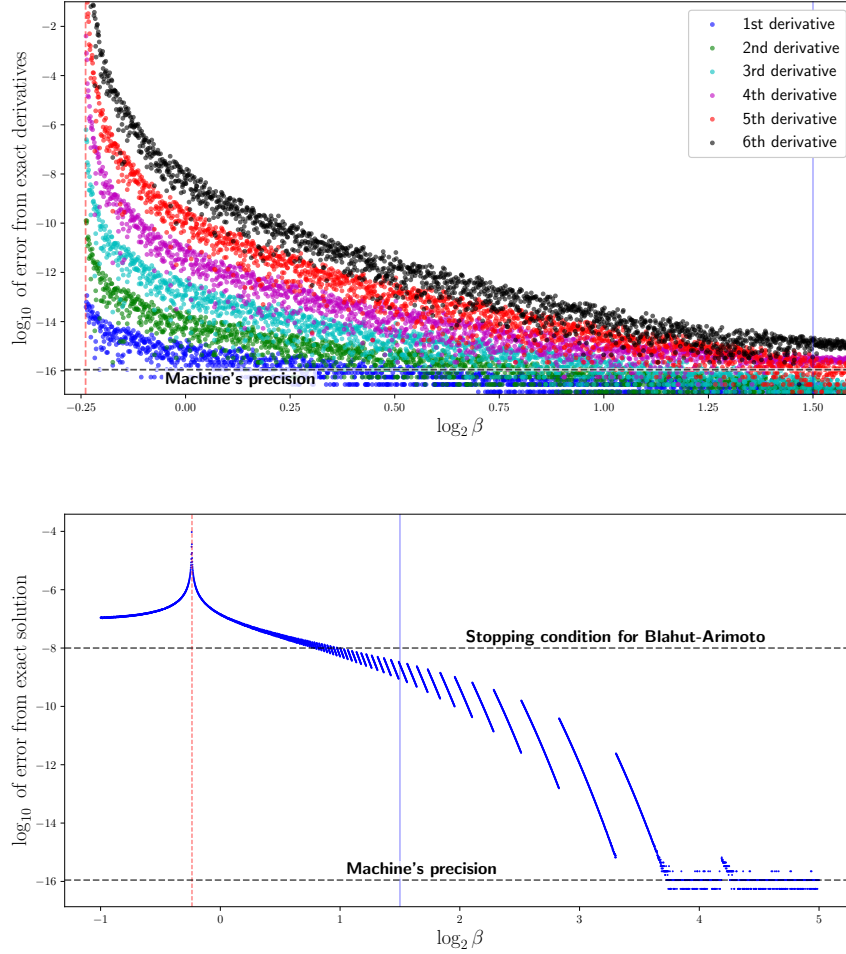
Figure 2.3: **Both Blahut-Arimoto and Algorithm 1 for RD can be remarkably accurate, yet lose their accuracy near a bifurcation** (dashed red verticals). To test the algorithms' accuracies, we compared their outputs to the analytical solutions of a binary source with Hamming distortion; see Section III.F for details. **Top**: Numerical derivatives were evaluated at the exact solutions using Algorithm 1 for RD. Their L^∞ distance to the analytical implicit derivatives is colored by order. As noted after Theorem 3, error due to numeric imprecision is amplified when approaching a bifurcation, by the derivatives' order. Only the range between the critical point β_c (F.6) of this problem and $\log_2 \beta = 1.5$ is shown at the top (red and blue verticals). Outside this region, the numerical derivatives' accuracy for this example is near the machine's precision. **Bottom**: L^∞-error of the solutions produced by Blahut-Arimoto, compared to the analytical solutions. A stopping condition of 10^{-8} between consecutive iterates in L^∞-norm was used, with uniform initial conditions for each β. This plot is explained by BA's critical slowing down, [Agmon et al., 2021] — see main text, after Theorem 4 in Section 2.3. Note that all four axes are in a logarithmic scale.

2.3 High-order derivative tensors of the Blahut-Arimoto operator

We provide closed-form formulae for the derivative tensors of the Blahut-Arimoto operator of arbitrary order. These are necessary to compute implicit derivatives at a root with Algorithm 1 for RD (in Section 2.2). Our main result, in Theorem 4, requires the derivatives to be evaluated at a marginal distribution of full support. This shall be dealt with when reconstructing the solution curve in Section 3.1, by *reducing* the RD problem to a smaller reproduction alphabet (see there).

As noted in Section 1, an RD problem is defined by a source distribution p_X, a reproduction alphabet $\hat{\mathcal{X}}$, and a distortion function $d(x, \hat{x})$. In the sequel, we shall often assume that the distortion is finite, and also *non-degenerate*: $d(\cdot, \hat{x}_1) \neq d(\cdot, \hat{x}_2)$ for all $\hat{x}_1 \neq \hat{x}_2$. Write $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$ for the source and reproduction alphabet sizes, respectively. When discussing rate-distortion problems, our notation shall differ from that used for root-tracking of an arbitrary operator F. Instead of $\boldsymbol{x}$, we write $\boldsymbol{r}$ for our variable, which now stands for a marginal distribution on $\hat{\mathcal{X}}$. That is, $F(\cdot, \beta)$ shall be an operator on $\mathbb{R}^M$ (previously $\mathbb{R}^T$). We further specialize F to capture fixed points of the Blahut-Arimoto algorithm below.

Recall that the Blahut-Arimoto algorithm (BA) Blahut [1972], Arimoto [1972] is defined by two equations; e.g., [Cover and Thomas, 2006, 10.8]. Given a marginal distribution $\boldsymbol{r}$ on the reproduction alphabet $\hat{\mathcal{X}}$, $r(\hat{x})$, define a test channel or *encoder* $\boldsymbol{q}$ by

$$q(\hat{x}|x) := \frac{r(\hat{x}) \, e^{-\beta d(x, \hat{x})}}{Z(x, \beta)} \, , \tag{2.18}$$

where $Z(x, \beta) := \sum_{\hat{x}'} r(\hat{x}') e^{-\beta d(x, \hat{x}')}$. Given an encoder $\boldsymbol{q}$, define a new marginal $\boldsymbol{s}$,

$$s(\hat{x}) := \sum_x p_X(x) q(\hat{x}|x) \tag{2.19}$$

A necessary (though *not* sufficient) condition for $\boldsymbol{r}$ to achieve the minimum $R(D)$ at (1.1) is that it is a fixed point of Equations (2.18) and (2.19), $\boldsymbol{s} = \boldsymbol{r}$. e.g., Berger [1971]. It is then called a *curve achieving distribution*, or simply an *achieving distribution*.

A curve-achieving marginal is determined by its test channel and vice-versa. So, a priori, it may not seem to matter which of the two is taken as the variable. However, it turns out (Section II.6.3) that only cluster-vanishing bifurcations can be detected in marginal coordinates. This shall be dealt with later, when reconstructing the solution curve (Assumption 3 in Section 3.1). For now, we define the BA_β operator as the composition of (2.18) followed by (2.19). It is a single iteration of the BA algorithm. Explicitly,

$$BA_\beta\left[\boldsymbol{r}\right](\hat{x}) := r(\hat{x}) \sum_x \frac{p_X(x) e^{-\beta d(x, \hat{x})}}{\sum_{\hat{x}'} r(\hat{x}') e^{-\beta d(x, \hat{x}')}} \tag{2.20}$$

is the operator's $\hat{x}$ coordinate when evaluated at $\boldsymbol{r}$. We could have taken instead the encoder $\boldsymbol{q}$ as our variable, defining $BA_\beta[\boldsymbol{q}]$ by the composition in reverse order, (2.19) followed by (2.18). However, to reduce computational costs, we select the lower-dimensional cluster marginal $\boldsymbol{r}$ as our variable.

Csiszár [1974] showed that the Blahut-Arimoto algorithm converges to an achieving distribution. An achieving distribution need not be unique, as Berger [1971] notes for example. Yet, it is a fixed point of BA_β (2.20), or equivalently a root of the operator $F := Id - BA_\beta$ (1.2), as Agmon et al. [2021] observed. Thus, to track fixed points of BA we calculate the derivative tensors of $Id - BA_\beta$ (1.2), below. See Section II.6 on its bifurcations. e.g., why should one expect RD problems to exhibit bifurcations, on types of RD bifurcations, and so forth. These shall be combined with implicit derivatives in RD when reconstructing the solution curve, in Section 3.1.

A few definitions are needed to write down the derivative tensors of $F := Id - BA_\beta$ (1.2). Define the real *polynomial ring on countably many variables* as the following set of finite sums,

$$\mathbb{R}[x_0, x_1, \dots] := \Big\{ \sum a_i \, x_{i_1}^{d_{i_1}} x_{i_2}^{d_{i_2}} \cdots x_{i_k}^{d_{i_k}} :$$

$$a_i \in \mathbb{R}, \ k \in \mathbb{N}, \ 0 \le i_1 < i_2 < \cdots < i_k, \ \text{and} \ d_{i_1}, d_{i_2}, \dots, d_{i_k} \in \mathbb{N}_0 \Big\} \quad (2.21)$$

where $\mathbb{N}$ stands for the natural numbers, and $\mathbb{N}_0$ for the non-negative integers. It consists of finite sums of monomials $x_{i_1}^{d_{i_1}} x_{i_2}^{d_{i_2}} \cdots x_{i_k}^{d_{i_k}}$ with real coefficients, with each $d_{i_j} \ge 0$. The *degree* of a polynomial in $\mathbb{R}[x_0, x_1, \dots]$ is the highest among its monomial degrees, which in turn is defined as $d_{i_1} + d_{i_2} + \cdots + d_{i_k}$ for the above. This slightly extends the usual definition of the (real) *polynomial ring* $\mathbb{R}[x_0, x_1, \dots, x_n]$ in multiple variables $x_0, x_1, \dots, x_n$. e.g., [Dummit and Foote, 2004, 9.1].

Next, define an $\mathbb{R}$-linear operator $\bar{d}$ on polynomials $\mathbb{R}[x_0, x_1, \dots]$ (2.21) by the Leibniz product rule $\bar{d}(f \cdot g) = g \cdot \bar{d}f + f \cdot \bar{d}g$. Except, that on the variables $x_0, x_1, \dots$, it is defined by

$$\bar{d}x_0 := 0, \quad \text{and} \quad \bar{d}x_k := x_1 \cdot x_k - x_{k+1} \quad \text{for } k > 0 , \quad (2.22)$$

and $\bar{d}c := 0$ for constant polynomials $c \in \mathbb{R}$. We note that the usual rules of polynomial differentiation such as $\bar{d}x_i^n = nx_i^{n-1}\bar{d}x_i$ follow from the Leibniz product rule. In algebraic context, a linear operator on a function space satisfying the Leibniz rule is known as a *derivation*. e.g., [Rotman, 2017, C-2.6].

With the operator $\bar{d}$ above, define as follows polynomials P_k in the $k+1$ variables $x_0, x_1, \dots, x_k$, for $k = 0, 1, 2, \dots$. First, set

$$P_0(x_0) := 1 . \quad (2.23)$$

Then, define P_{k+1} inductively in terms of P_k,

$$P_{k+1}(x_0, x_1, \dots, x_{k+1}) := (x_1 - x_0) \cdot P_k + \bar{d}P_k \quad (2.24)$$

With this, the first few polynomials are seen (in Section II.5.3) to be

$$P_1(x_0, x_1) = x_1 - x_0 \quad (2.25)$$

$$P_2(x_0, x_1, x_2) = x_0^2 - 2x_0 x_1 + 2x_1^2 - x_2 \quad (2.26)$$

$$P_3(x_0, x_1, x_2, x_3) = -x_0^3 + 3x_0^2 x_1 + 3x_0 x_2 - 6x_0 x_1^2 + 6x_1^3 - 6x_1 x_2 + x_3 \quad (2.27)$$

For an encoder $\boldsymbol{q}$ and a distortion $d(x, \hat{x})$, denote

$$\mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')^k\right] := \sum_{\hat{x}'} q(\hat{x}'|x) d(x, \hat{x}')^k \quad (2.28)$$

for $k > 0$ and a particular coordinate $x \in \mathcal{X}$. By abuse of notation, define for $k \ge 0$,

$$P_k[\boldsymbol{q}; d](\hat{x}, x) := P_k\Big(d(x, \hat{x}), \mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')\right], \dots, \mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')^k\right]\Big) \quad (2.29)$$

That is, $P_k[\boldsymbol{q}; d]$ is a function of two variables $\hat{x}$ and x, defined by a pointwise evaluation of the polynomials P_k (2.23)-(2.24) at $x_0 := d(x, \hat{x})$ and $x_k := \mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')^k\right]$ for $k > 0$. We shall write $P_k(\hat{x}, x)$ for short when the distortion $d(x, \hat{x})$ and the point $\boldsymbol{q}$ of evaluation are understood.

We are now set to spell out the derivative tensors of $(Id - BA_\beta)[\boldsymbol{r}]$ (1.2). While the entries of $(Id - BA_\beta)[\boldsymbol{r}]$ are indexed by $\hat{x}$, we index the entries of its derivative tensors by a multi-index $\boldsymbol{\alpha} \in \mathbb{N}_0^{M+1}$, for convenience. See comments on indexation after definition (2.6) of a derivatives tensor $D_{\beta^b, \boldsymbol{x}^m}^{b+m} F$ (in Section 2.2).

Theorem 4 (High-order partial derivatives of $Id - BA_\beta$ (1.2)). *Let p_X and $d(x, \hat{x})$ define an RD problem on the reproduction alphabet $\hat{\mathcal{X}}$. Let $\boldsymbol{r} \in \Delta[\hat{\mathcal{X}}]$, and let $\boldsymbol{q}$ be the encoder defined by it via Equation (2.18). Then, for any integer $\alpha_0 > 0$,*

$$\frac{\partial^{\alpha_0}}{\partial \beta^{\alpha_0}} \big(Id - BA_\beta\big)[\boldsymbol{r}](\hat{x}) = -\sum_x p_X(x) q(\hat{x}|x) P_{\alpha_0}(\hat{x}, x) \,, \tag{2.30}$$

where $P_{\alpha_0}(\hat{x}, x)$ is defined by (2.29).

Assume further that $\boldsymbol{r}$ is of full support, $r(\hat{x}) > 0$ for every $\hat{x}$. Let $\boldsymbol{\alpha} = (\alpha_0, \boldsymbol{\alpha}_+) \in \mathbb{N}_0^{M+1}$ be a multi-index with $\boldsymbol{\alpha}_+ \neq \boldsymbol{0}$. Then,

$$\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} \left(Id - BA_\beta\right)[\boldsymbol{r}](\hat{x})$$

$$= \delta_{\boldsymbol{\alpha}, \boldsymbol{e}_{\hat{x}}} - (-1)^{|\boldsymbol{\alpha}_+| - 1}(|\boldsymbol{\alpha}_+| - 1)! \, \boldsymbol{\alpha}! \sum_x p_X(x) \left(\frac{q(\hat{x}'|x)}{r(\hat{x}')}\right)^{\boldsymbol{\alpha}_+} \sum_{\boldsymbol{k} \in \mathbb{N}_0^M : \, |\boldsymbol{k}| = \alpha_0} \left(\prod_{i \neq \hat{x}} G\big(k_i, \alpha_i; \boldsymbol{q}, d\big)_{(\hat{x}_i, x)}\right)$$

$$\cdot \left[\alpha_{\hat{x}} \cdot G\big(k_{\hat{x}}, \alpha_{\hat{x}}; \boldsymbol{q}, d\big)_{(\hat{x}, x)} - |\boldsymbol{\alpha}_+| \cdot (1 + \alpha_{\hat{x}}) \cdot q(\hat{x}|x) \cdot G\big(k_{\hat{x}}, 1 + \alpha_{\hat{x}}; \boldsymbol{q}, d\big)_{(\hat{x}, x)}\right] \tag{2.31}$$

where $G\big(k, a; \boldsymbol{q}, d\big)$ is defined on integers $k, a \geq 0$ by $G = 0$ if $a = 0 < k$, and otherwise

$$G\big(k, a; \boldsymbol{q}, d\big)_{(\hat{x}, x)} := \sum_{\substack{\boldsymbol{t} \in \mathbb{N}_0^k : \, |\boldsymbol{t}| \leq a, \\ \sum_j j \cdot t_j = k}} \frac{1}{\boldsymbol{t}! \, (a - |\boldsymbol{t}|)!} \prod_{j=1}^{k} \left(\frac{P_j(\hat{x}, x)}{j!}\right)^{t_j} \,. \tag{2.32}$$

With the above derivative tensors of $Id - BA_\beta$ (1.2), Algorithm 1 for computing higher implicit derivatives (in Section 2.2) can now be specialized to RD. This allows us to compute the implicit derivatives $\frac{d^l \boldsymbol{r}}{d\beta^l}$ (1.4) at an RD root $\boldsymbol{r}$. The Theorem's proof is outlined in Section II.5. In addition to the RD ODE (Theorem 14), the main results there are the formulas for the encoder's repeated partial β-derivatives and its mixed ones (Propositions 12 and 15), which yield (2.30) and (2.31). Section II.5.5 comments how the latter can be computed efficiently.

The implicit derivatives computed by Algorithm 1 for RD lose their numerical stability when approaching a bifurcation, losing it faster if the order is higher (see after Theorem 3 in Section 2.2). Other than that, the first few implicit derivatives computed for RD are seen in Figure 2.3 (top) to be remarkably accurate. Further, note that the derivative tensors of $Id - BA_\beta$ might be numerically unstable when computed for high orders, due to the factorials at the denominators of (2.32). Interestingly, the bifurcation's presence also affects the classic Blahut-Arimoto algorithm, Blahut [1972], Arimoto [1972]. It too suffers from accuracy loss near bifurcations, as demonstrated by Figure 2.3 (bottom). This can be seen as a direct consequence of the critical slowing down near RD bifurcations, Agmon et al. [2021].

3 Reconstructing an RD solution curve from implicit derivatives

In Section 2, we showed how to compute implicit derivatives at an RD root, using the specialization of Algorithm 1 to RD. Now, we shall leverage these derivatives to reconstruct the entire solution curve of a given RD problem.

In Section 3.1, we modify the vanilla fixed-step Taylor method to track an RD root numerically until a cluster-vanishing bifurcation is detected — which is Algorithm 2. We handle these bifurcations in Algorithm 3 (in Section 3.2), thereby reconstructing the entire solution curve, as

seen in Figure 3.1. The algorithms' error from the true solution is of order $O(|\Delta\beta|^l)$ (Theorem 5), for an l-th order Taylor method of step $\Delta\beta$. Both Blahut-Arimoto and the implicit numerical derivatives we use suffer from degraded accuracy near RD bifurcations, as shown earlier in Figure 2.3. However, while BA suffers from a hefty computational penalty near bifurcations due to its critical slowing down, [Agmon et al., 2021], the added computational cost to our Algorithms 2 and 3 is negligible.
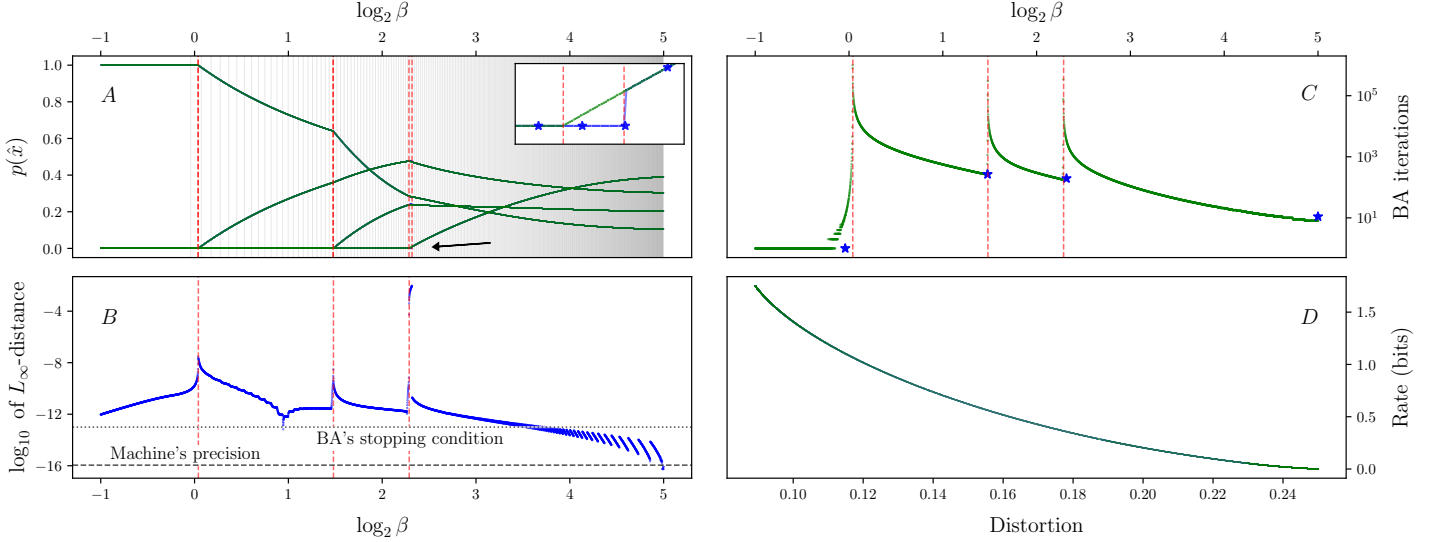

Figure 3.1: **Reconstructing a solution curve from implicit derivatives, with root-tracking for RD (Algorithm 3).** Reproducing the problem in [Agmon et al., 2021, Figure 1], defined by
$d(x,\hat{x}) = \frac{1}{8}\begin{pmatrix} 0 & 1 & 1 & 2 \\ 4 & 1 & 5 & 2 \\ 4 & 5 & 1 & 2 \\ 8 & 5 & 5 & 2 \end{pmatrix}$ and $p_X = \frac{1}{10}(4,3,2,1)$. Bifurcations are marked by dashed red verticals.
A. Cluster marginal $p(\hat{x})$ as a function of β. A fixed-width grid of almost 400 points is selected along the β-axis (gray verticals), with implicit derivatives computed to the 7th order at each grid-point. Every point is extrapolated from the grid-point to its right using a modified Taylor method (Algorithm 2 in Section 3.1), except for the rightmost one and near the bifurcations, where Blahut-Arimoto's algorithm is used. The cluster vanishing bifurcations of this problem are handled by Algorithm 3 (with $\delta = 10^{-2}$, $\Delta\beta = {-31.5}/{400}$), which can either overshoot or undershoot a bifurcation (Section 3.2). On undershooting, the bifurcation is detected too early (redundant red vertical to the right), and the algorithm switches temporarily to a sub-optimal solution branch as a result. This is magnified $\times 20$ in the top-right inset, depicting an undershooting of the rightmost bifurcation, with four nearby grid-points marked by blue stars. The entire solution curve (in blue) is extrapolated from the grid. For comparison, Blahut-Arimoto's is in green (a 10^{-13} stopping condition, 5000 grid-points). The two are visually indistinguishable almost everywhere. **B.** The L_∞-distance between the solutions produced by BA and Algorithm 3 (this problem has no analytic solution). Note the localized error in Algorithm 3 due to switching to the sub-optimal branch near the rightmost bifurcation. **C.** Number of Blahut-Arimoto iterations. BA in reverse annealing is in green; Algorithm 3 invokes BA only four times (blue stars): once for the initial condition, and once per bifurcation. Note that Algorithm 3 avoids the critical slowing down exhibited by BA, Agmon et al. [2021]; see Section 3.2. **D.** The rate-distortion curve of this problem. The results of BA and Algorithm 3 are indistinguishable.

Bounds on the algorithms' computational and memory complexities are provided in Section 3.3. One might expect the computational cost to increase if a smaller error is required. Indeed, we estimate of this tradeoff for our algorithms. Unlike the Blahut-Arimoto algorithm, which computes solutions only at specified grid points, when our root-tracking Algorithm 3 terminates, only a fixed computational cost is needed to extrapolate any off-grid point. In principle, the entire solution curve can be approximated from few grid points by using high-order expansions, at least in well-behaved examples. Although high orders may not be practical due to their computational costs, this is demonstrated in Figure 3.2. Suggestions on improving our algorithms' efficiency are discussed in Section 3.4.

This section is complemented by several Sections in Part II, corresponding to the subsections of Section 3. In Section II.6, we provide several basic results on RD bifurcations, to our knowledge for the first time. This allows us to tell when does our Algorithm 3 follow the optimal branch, explain why it may fail, and consider improvements. Section II.7 follows standard error analysis results in the literature to provide the guarantees and error estimates for our case. In the context of numerical solutions of ODEs, a computational difficulty stems from the existence of bifurcations. Thus, in a sense, our algorithm trades the bifurcations' hefty computational cost due to Blahut-Arimoto's critical slowing down with reduced accuracy near bifurcations. Finally, Section II.8 bounds the computational and memory costs, both of root-tracking in general, and of its specialization to rate-distortion.

3.1 A modified Taylor method for RD root-tracking

We parameterize a root of $Id - BA_\beta$ (1.2) by its cluster marginal $r(\hat{x})$ rather than by the encoder $q(\hat{x}|x)$ to reduce computational costs, as noted in Section 2.3. Write $\tilde{\boldsymbol{r}}_{\beta_n}$ or simply $\tilde{\boldsymbol{r}}_n$ for a numerical approximation of the true solution $\boldsymbol{r}_{\beta_n} \in \Delta[\hat{\mathcal{X}}]$ at β_n. Starting at $\tilde{\boldsymbol{r}}_0 := \boldsymbol{r}_{\beta_0}$, set[5]

$$\tilde{\boldsymbol{r}}_{n+1} := \tilde{\boldsymbol{r}}_n + \Delta\beta \cdot T_l\big(\tilde{\boldsymbol{r}}_n, \beta_n, \Delta\beta\big) . \tag{3.1}$$

Where, $T_l(\boldsymbol{x}, \beta, \Delta\beta) := \frac{d\boldsymbol{x}}{d\beta}\big|_{(\boldsymbol{x},\beta)} + \frac{1}{2!}\frac{d^2\boldsymbol{x}}{d\beta^2}\big|_{(\boldsymbol{x},\beta)}\Delta\beta + \cdots + \frac{1}{l!}\frac{d^l\boldsymbol{x}}{d\beta^l}\big|_{(\boldsymbol{x},\beta)}\Delta\beta^{l-1}$ is the l-th order Taylor remainder as in expansion (1.5), with the derivatives evaluated by Algorithm 1 for RD (Section 2.2) at $(\tilde{\boldsymbol{r}}_n, \beta_n)$. We extrapolate off-grid points $\tilde{\boldsymbol{r}}_\beta$ from (3.1) using the last grid point, except in some cases near a bifurcation (see Section 3.2). This numerical method is known as the *Taylor method*; e.g., Atkinson et al. [2011], Butcher [2016]. See Section II.7 for a recap and its error analysis. While there are other numerical methods that use derivatives to approximate $\boldsymbol{r}_\beta$, we chose the Taylor method since it is simple and well-studied. cf., Section 3.4 on improvements.

Computing implicit derivatives with Algorithm 1 for RD requires the derivative tensors of BA_β, which were provided by Theorem 4 (Section 2.3). The formulae there require a marginal $\boldsymbol{r}$ to be of full support. To handle this and to further reduce computational costs, define the following. For a proper nonempty subset $\hat{\mathcal{X}}'$ of the reproduction alphabet $\hat{\mathcal{X}}$, define *the RD problem restricted to $\hat{\mathcal{X}}'$* by deleting letters $\hat{x} \in \hat{\mathcal{X}}$ outside $\hat{\mathcal{X}}'$ and the respective columns $d(\cdot, \hat{x})$ from the distortion matrix. For practical purposes, deleting a letter $\hat{x}$ is equivalent to allowing initializations $\boldsymbol{r}_0$ of BA only if their $\hat{x}$ coordinate is zero, $r_0(\hat{x}) = 0$, as can be seen by the explicit form (2.20) of BA_β. When the marginal $\boldsymbol{r}$ is understood, we call the RD problem restricted to supp $\boldsymbol{r}$ *the reduced problem*. Reducing a problem does not affect the solution, [Berger, 1971, Lemma 1 in Chapter 2].

We require henceforth that $d(x, \hat{x})$ is finite and non-degenerate[6]. By definition, $\boldsymbol{r}$ is of full support in the reduced problem. Therefore, the Jacobian $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}}$ is non-singular in the reduced problem, [Agmon et al., 2021, Theorem 1 ff.], when $\boldsymbol{r}$ is also a fixed point of BA_β.

---

[5]We note that the approximations $\tilde{\boldsymbol{r}}_n$ (3.1) need *not* be normalized distributions. However, Theorem 5 below guarantees that $\tilde{\boldsymbol{r}}_n$ does not deviate much from the true solution $\boldsymbol{r}_{\beta_n}$ when the step size $|\Delta\beta|$ is small enough. Thus, $\tilde{\boldsymbol{r}}_n$ does not deviate much from being normalized. Note also the normalization on step 8 of Algorithm 3.

[6]Namely, the ($\hat{x}$ indexed) columns of the distortion matrix d are distinct — see Section 2.3.

In particular, the numerical derivatives $\frac{d^l \boldsymbol{r}}{d\beta^l}$ produced by Algorithm 1 for RD are then defined uniquely, as explained after Theorem 3 (in Section 2.2). These are vectors of a lower dimension $|\operatorname{supp} \boldsymbol{r}| \leq M$, tracing the root's path in the reduced problem; optimality of this path is discussed below. The explicit form of $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}}$ at an arbitrary distribution $\boldsymbol{r}$ is given later, by Corollary 10 (Section II.5.2). Its formula (5.7) shows that the Jacobian is non-singular even when straying slightly off a stable fixed point of BA_β (e.g., due to accumulated approximation error).

The argument at Agmon et al. [2021] further shows that an eigenvalue of $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}}$ vanishes if and only if $r(\hat{x})$ vanishes for some $\hat{x}$. This gives a simple way to detect cluster-vanishing bifurcations. Namely, an RD bifurcation where $r(\hat{x})$ gradually vanishes. We take only negative steps $\Delta\beta < 0$ when considering how fixed points evolve — see Section II.6.2 ff. As shown there, cluster-vanishing bifurcations are indeed bifurcations, where two roots collide and merge. To detect these, one can set a cluster-mass threshold $\delta > 0$ below which $r(\hat{x})$ is considered to have vanished. Thus, iterating over a Taylor method step (3.1) until a cluster vanishes allows to reconstruct the root's path till the next bifurcation. This is summarized by Algorithm 2, with convergence guarantees in Theorem 5 below. We handle the bifurcation later, in Section 3.2. A significant part of the computational difficulty stems from approaching a bifurcation (Section II.7.2). Thus, setting a threshold $\delta > 0$ on the cluster mass effectively restricts the problem's difficulty.

Algorithm 2: A modified l-th order Taylor method, tracking an RD root to a cluster-vanishing bifurcation

---

1: **function** TRACK RD ROOT TO BIFURCATION($\boldsymbol{r}_{\beta_0}, \beta_0; \Delta\beta, \delta, l, d, p_X$)

Input:
   β_0, a root $\boldsymbol{r}_{\beta_0}$ of $Id - BA_{\beta_0}$ (1.2) of full support, step-size $\Delta\beta < 0$,
   cluster mass threshold $\delta > 0$, an order $l > 0$, and an RD problem (d, p_X).

Output:
   Approximations $\tilde{\boldsymbol{r}}_i$ of the true solution $\boldsymbol{r}_{\beta_i}$ at β_i,
   and the l-th order Taylor expansions p_i around $(\tilde{\boldsymbol{r}}_i, \beta_i)$.

2:     Initialize $\tilde{\boldsymbol{r}}_0 \leftarrow \boldsymbol{r}_{\beta_0}, n \leftarrow 0$.

3:     **while** $\forall \hat{x}' \; \tilde{r}_n(\hat{x}') > \delta$ **and** $\beta_n > 0$ **do**   ▷ Stop if too close to a bifurcation.

4:         **for** $i = 1, \ldots, l$ **do**

5:             $\left.\frac{d^i \boldsymbol{r}}{d\beta^i}\right|_{(\tilde{\boldsymbol{r}}_n, \beta_n)} \leftarrow$ CALCULATE IMPLICIT DERIVATIVE($\tilde{\boldsymbol{r}}_n, \beta_n, i; d, p_X$)

                                                                      ▷ Algorithm 1 for RD.

6:         **end for**

7:         $p_n(\Delta) \leftarrow \tilde{\boldsymbol{r}}_n + \frac{\Delta}{1!} \cdot \left.\frac{d\boldsymbol{r}}{d\beta}\right|_{(\tilde{\boldsymbol{r}}_n, \beta_n)} + \cdots + \frac{\Delta^l}{l!} \cdot \left.\frac{d^l \boldsymbol{r}}{d\beta^l}\right|_{(\tilde{\boldsymbol{r}}_n, \beta_n)}$

8:         $\tilde{\boldsymbol{r}}_{n+1} \leftarrow p_n(\Delta\beta)$

9:         $\beta_{n+1} \leftarrow \beta_n + \Delta\beta$

10:        $n \leftarrow n + 1$

11:    **end while**

12:    **return** $\{(\tilde{\boldsymbol{r}}_i, \beta_i, p_i)\}_{i=1}^n$ ▷ Approximations of the true solutions $\boldsymbol{r}(\beta_n)$.

13: **end function**

Roots $\boldsymbol{r}_\beta$ of an RD problem are very well-behaved — namely, piecewise analytic in β. Reduction of an RD problem mods out the kernel of the Jacobian in cluster-marginal coordinates $\boldsymbol{r}$, leaving it non-singular, and yet is simple and straightforward to implement. By the implicit function theorem [Kielhöfer, 2012, (I.1.7)], there not only exists a unique root $\boldsymbol{r}_\beta$ of the reduced problem (through a given $\boldsymbol{r}_{\beta_0}$), but it is also (real) analytic. For, the BA_β (2.20) operator is a composition of analytic functions. Recall, e.g., [Dieudonné, 1969, IX.3], that $\boldsymbol{r}_\beta$ is *analytic* (in some open set) if its infinite-order Taylor expansion (1.5) about any β_0 not only exists ($\boldsymbol{r}_\beta$ is smooth), but also converges pointwise to the function itself, within some radius of convergence. Every coordinate

$r_\beta(\hat{x})$ of $\boldsymbol{r}_\beta$ is nothing but an (infinite) power series in $\Delta\beta := \beta - \beta_0$. The partial sums of a power series converge uniformly on closed intervals inside their radius of convergence. That is, only a finite number of summands is needed to extrapolate the solution on an interval, at a given precision. In principle, this allows us to extrapolate the entire solution curve by using Taylor expansions at just a few points if the convergence radii are large enough, as demonstrated by Figure 3.2.
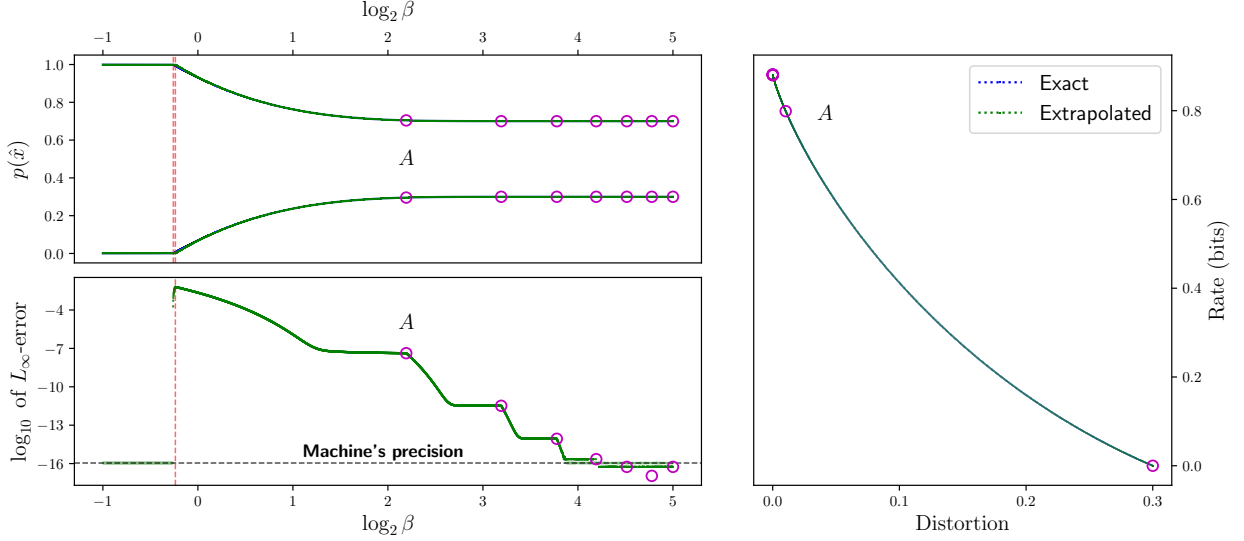

Figure 3.2: **Extrapolating the entire solution curve from 7 grid points** (magenta circles), **using Algorithm 3 of order 20**, for Binary source with Hamming distortion III.F, with $\delta = 10^{-2}$. Although the error guarantees of Theorem 5 aim at small step-sizes, and high-order derivatives are computationally expensive, this example demonstrates the predictive power achievable by exploiting real-analyticity in well-behaved examples; see comments in the main text.
Top left: The extrapolations (in green) from the grid points overlap the exact solutions (in blue) almost precisely, deviating visibly only near the bifurcation at $\log_2 \beta_c \approx -0.24$. The algorithm overshoots the bifurcation slightly (Section 3.2), marked by an extra dashed red vertical to the left. Two Blahut-Arimoto iterations were needed to compute the rightmost grid point, with the next ones extrapolated using a 20th-order Taylor polynomial with a fixed step size of $\Delta\beta \approx -4.57$.
Bottom left: L_∞-error between the extrapolation generated from Algorithm 3 and the analytical solution of this problem, in Section III.F. For the most part, the achieved error is near the machine's precision (note the logarithmic scale of the horizontal axis), increasing notably near the bifurcation. While the point of bifurcation is an essential difficulty, Section II.7, the decreased accuracy to the left of point A could be improved upon, e.g., by varying the step-size; see Section 3.4. **Right**: The problem's exact RD curve is visually indistinguishable from the extrapolated one. While the first few grid points cluster to the top-left, most of the curve happens to be extrapolated from the single point A.

While the Jacobian $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}}$ in marginal coordinates $\boldsymbol{r}$ provides an appealing picture of piecewise analytic roots, which alter their course only when colliding at the simplex boundary, it does not tell the whole story. For, the Jacobian in these coordinates can detect only cluster-vanishing bifurcations, as discussed later in Section II.6.3. Indeed, RD problems have a plethora of sub-optimal solutions (Section II.6.1). These may, for example, exchange optimality in another kind of bifurcation which we call *support-switching*, which is demonstrated by Figure 6.2 (in Section II.6). We show later (Section II.6.5) that these can explain the linear segments seen in RD curves

[Berger, 1971], at least in some cases, as well as some instances of critical slowing down which are not explained by Agmon et al. [2021] (see Section II.6.3 and Figure 6.2).

For our purposes, we would like to ensure that the root tracked by our algorithms remains optimal (an achieving distribution), rather than a sub-optimal one. This is done in two steps. First, ensure that Algorithm 2 starts at an achieving distribution $\boldsymbol{r}_{\beta_0}$, by invoking Blahut-Arimoto until convergence to an initial condition of full support, Csiszár [1974] (brought as Theorem 17 in Section II.6.1). Second, so long that Algorithm 2 starts at an achieving distribution, then the root it tracks remains optimal. Similarly, for the below Algorithm 3 (Section 3.2). This follows from Assumption 3 below, as we show later (in Section II.6.4).

At a fixed point of BA_β (2.20), the cluster marginal $\boldsymbol{r}_\beta$ is determined by an encoder $\boldsymbol{q}_\beta$ via Equation (2.19), and vice versa (2.18). However, the MN-by-MN Jacobian matrix $D_{\boldsymbol{q}}(Id - BA_\beta)\big|_{\boldsymbol{q}_\beta}$ with respect to the encoder $\boldsymbol{q}$ contains more information than the M-by-M Jacobian $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}_\beta}$ with respect to the marginal $\boldsymbol{r}$ — see Section II.6.3. Where, recall that $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$. Indeed, the $\boldsymbol{q}$-Jacobian can detect RD bifurcations of any kind, unlike the $\boldsymbol{r}$-Jacobian; see Corollary 21 (in Section II.6.3), and Proposition 23 for its explicit form. Therefore, we require

Assumption 3. *The $\boldsymbol{q}$-Jacobian $D_{\boldsymbol{q}}(Id - BA_\beta)\big|_{\boldsymbol{q}_\beta}$ is non-singular when evaluated at achieving distributions $\boldsymbol{q}_\beta$, except at cluster-vanishing bifurcations.*

This assumption guarantees that the RD problem is well-behaved (e.g., non-degenerate), and that it has a unique optimal solution whose path undergoes only cluster vanishing bifurcations[7]. It holds for most of the examples in this paper[8]. In particular, the earlier Assumptions 1 and 2 (of Section 2.1) necessary for calculating implicit derivatives follow from Assumption 3 (see Section II.6.4). The $\boldsymbol{q}$-Jacobian $D_{\boldsymbol{q}}(Id - BA_\beta)\big|_{\boldsymbol{q}_\beta}$ is singular if the distortion matrix is degenerate (Section II.6.3). However, other than that, we find it reasonable to require that the $\boldsymbol{q}$-Jacobian is non-singular outside of cluster-vanishing bifurcations (Assumption 3). Indeed, [Berger, 1971, Chapter 2] says that "usually, each point on the rate-distortion curve... is achieved by a unique conditional probability assignment. However, if the distortion matrix exhibits certain form of symmetry and degeneracy, there can be many choices of [a minimizer]."

One can test Assumption 3 directly, by calculating the eigenvalues of the $\boldsymbol{q}$-Jacobian periodically. These are expected to vanish only if some cluster vanishes simultaneously — see Equation (6.6) for details (Section II.6.3). While our algorithms can be extended to handle RD bifurcations also of other kinds, using the bifurcations Section II.6, it is beyond the scope of this work.

We conclude this subsection with convergence guarantees for our modified Taylor method at Algorithm 2 — see its error analysis in Section II.7. For $\delta > 0$, the *closed δ-interior* of the simplex $\Delta[\hat{\mathcal{X}}]$ consists of the distributions $\boldsymbol{r} \in \Delta[\hat{\mathcal{X}}]$ with $r(\hat{x}') \geq \delta$ for all $\hat{x}' \in \hat{\mathcal{X}}$. Note that this is not the same as the interior of the simplex, which is an open set.

Theorem 5 (Taylor method converges uniformly between RD bifurcations, on a full support)**.** *Let $\boldsymbol{r}_\beta$ be a root of the RD problem defined by p_X and a finite non-degenerate distortion measure d. Suppose that $\boldsymbol{r}_\beta$ is of full support at $\beta_0 > 0$. Let $\delta > 0$ be a cluster mass threshold, such that $\delta < \min_{\hat{x}} r_{\beta_0}(\hat{x})$. Then there exists $0 \leq \beta_f(\delta) < \beta_0$ such that,*

1. *$\boldsymbol{r}_\beta$ is in the closed δ-interior of $\Delta[\hat{\mathcal{X}}]$ for $\beta \in [\beta_f(\delta), \beta_0]$; and*

2. *For $l > 0$, the error of an l-th order Taylor method satisfies*

$$\max_{\beta \in [\beta_f(\delta), \beta_0]} \|\tilde{\boldsymbol{r}}_\beta - \boldsymbol{r}_\beta\|_\infty = O(|\Delta\beta|^l) \tag{3.2}$$

---

[7]This does not rule out the possibility that sub-optimal roots undergo other kinds of bifurcations.
[8]Except for the right bifurcation in Figure 6.2, at Section II.6.

for $|\Delta\beta| > 0$ small enough, and $\tilde{r}_\beta$ the Taylor method approximations defined by (3.1).

Since Algorithm 2 tries to stop δ-away from a bifurcation, its error converges uniformly for small enough step sizes, at a rate (3.2) proportional to its order l. All the RD roots we compute are generated by Algorithm 2, except for the initial point and those too close to a bifurcation. Bifurcations are handled by Algorithm 3 below (Section 3.2). It will reduce the problem at hand, guaranteeing the full support required by Theorem 5. Thus, we obtain an error guarantee for nearly all the generated grid points. The proof of Theorem 5 (in Section III.D.2) is based on standard Taylor method error analysis, brought at Theorem 25 (in Section II.7.1). Its crux is that implicit derivatives and the local Lipschitz constants of Taylor's method are bounded uniformly, on suitable compact subsets in the simplex interior; see Sections II.7.2-7.3.

The convergence guarantees of Theorem 5 suggest considerations for selecting the parameters of Algorithm 2. The cluster mass threshold $\delta > 0$ obviously should not be too large, so that a bifurcation is not accidentally detected when there is none. On the other hand, increasing δ restricts the algorithm's computational difficulty, as discussed in Section II.7. One would then like to select $|\Delta\beta|$ small enough and the order l large enough such that Equation (3.2) (in Theorem 5) guarantees that $\tilde{r}_\beta$ does not deviate too much from the true solution r_β. This guarantees the solution's accuracy and ensures that comparing a cluster mass $\tilde{r}_\beta(\hat{x})$ against the threshold δ is meaningful (on line 3 of Algorithm 2). On the other hand, decreasing $|\Delta\beta|$ or increasing l impacts the algorithms' complexities, as elaborated in Section 3.3. See the Figures throughout and our implementation for sample values. With that, the considerations implied by Theorem 5 are not the only way to choose the algorithms' parameters. For example, motivated by the earlier discussion on analyticity, Figure 3.2 depicts high-quality approximations even though its step size $\Delta\beta$ is large.

3.2 RD root-tracking near bifurcations

We proceed with the solution's handling where Algorithm 2 left off, once a cluster-vanishing bifurcation has been detected. The following heuristic re-gains accuracy while avoiding the computational cost due to Blahut-Arimoto's critical slowing down near the bifurcation, [Agmon et al., 2021]. Assumption 3 is required below.
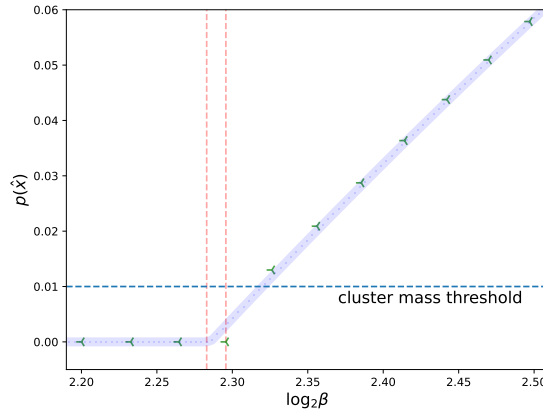
Figure 3.3: **Undershooting a bifurcation**. The bifurcation is detected early (right dashed red vertical), slightly to the right of its true position (left vertical). Once the threshold is crossed, our heuristic zeros the nearly-vanished cluster (at right vertical, not shown). The distribution it yields (at right vertical, marked) achieves a sub-optimal branch (not shown), in which this symbol is zero. However, optimality is re-gained shortly afterwards, when reaching the bifurcation; see main text. The markers denote the output of Algorithm 3 of second-order on a 300-point grid, with BA for comparison in blue, near the rightmost bifurcation of Figure 3.1.

Consider the first approximation $(\tilde{\boldsymbol{r}}_n, \beta_n)$ after the cluster mass threshold δ has been crossed. We zero any cluster $\hat{x}$ for which $\tilde{r}_n(\hat{x}) < \delta$, normalize, and then use the resulting distribution $\tilde{\boldsymbol{r}}'$ as the initial condition for Blahut-Arimoto's algorithm. $\tilde{\boldsymbol{r}}'$ has precisely those nearly-vanished clusters set to zero. Thus, Blahut-Arimoto converges to an achieving distribution $\tilde{\boldsymbol{r}}_n''$ of the problem reduced to $\operatorname{supp} \tilde{\boldsymbol{r}}'$, [Csiszár, 1974] (see Section II.6.1). We then invoke Algorithm 2 anew on the reduced problem, starting at the achieving distribution $(\tilde{\boldsymbol{r}}_n'', \beta_n)$. This process repeats so long that $\beta_n > 0$ and the initial condition $\tilde{\boldsymbol{r}}_n''$ is non-trivial. We summarize this heuristic in Algorithm 3.

The first approximation $(\tilde{\boldsymbol{r}}_n, \beta_n)$ after crossing the threshold may either be earlier than[9] the true point of bifurcation β_c, $\beta_c < \beta_n$, called an *undershooting*. Or the bifurcation may be detected too late, $\beta_n \leq \beta_c$, an *overshooting*. Either of these must happen when a bifurcation is detected, usually with strict inequality. A sub-optimal root exists at a right vicinity of a bifurcation, $\beta > \beta_c$, as shown in Section II.6.2. It is a solution of the reduced problem to $\operatorname{supp} \tilde{\boldsymbol{r}}'$, with $\tilde{\boldsymbol{r}}'$ the normalized zeroization as above. The optimal and sub-optimal roots intersect at β_c, merging there into one. On an overshoot, $\beta_n \leq \beta_c$, the initial condition $\tilde{\boldsymbol{r}}_n''$ produced by the heuristic will remain on the optimal branch. On an undershoot, $\beta_c < \beta_n$, $\tilde{\boldsymbol{r}}_n''$ will be on the sub-optimal branch of smaller support $\operatorname{supp} \tilde{\boldsymbol{r}}'$ while re-gaining optimality shortly thereafter, as demonstrated by Figure 3.3.

Invoking Blahut-Arimoto before starting anew with Algorithm 2 re-gains accuracy, compensating for the approximation errors accumulated so far, while also ensuring that the root being tracked is optimal on the non-vanished clusters. Further, this allows us to *avoid* Blahut-Arimoto's critical slowing down. For, as shown by Agmon et al. [2021], the latter is exhibited due to a Jacobian eigenvalue that gradually vanishes when approaching a bifurcation. This eigenvalue corresponds to a cluster of vanishing marginal and so is removed once that cluster is removed, before invoking Blahut-Arimoto on line 9 of Algorithm 3. The lack of critical slowing down is consistent with our numerical results.

Algorithm 3: Root-tracking for RD, with l-th order Taylor method.

Input:
> β_0, a root $\boldsymbol{r}_{\beta_0}$ of $Id - BA_{\beta_0}$ (1.2), a step-size $\Delta\beta < 0$, a cluster mass threshold $0 < \delta < 1$,
> an order $l > 0$, and an RD problem (d, p_X).

Output:
> Approximations $\tilde{\boldsymbol{r}}_i$ of the true solution $\boldsymbol{r}_{\beta_i}$ at β_i,
> and the l-th order Taylor expansions p_i around $(\tilde{\boldsymbol{r}}_i, \beta_i)$.

1: Initialize $\tilde{\boldsymbol{r}} \leftarrow \boldsymbol{r}_{\beta_0}, \beta \leftarrow \beta_0, results \leftarrow \{\}$.
2: **while** $|\operatorname{supp} \tilde{\boldsymbol{r}}| > 1$ **and** $\beta > 0$ **do**   ▷ Stop if solution has a trivial support.
3:     $\tilde{\boldsymbol{r}}, \bar{d} \leftarrow$ the reduction of $\tilde{\boldsymbol{r}}, d$ to $\operatorname{supp} \tilde{\boldsymbol{r}}$.
4:     *solution path* $\leftarrow$ TRACK RD ROOT TO BIFURCATION$(\tilde{\boldsymbol{r}}, \beta; \Delta\beta, \delta, l, \bar{d}, p_X)$.

> ▷ Algorithm 2.

5:     Append items in *solution path* to *results*, except for the last.
6:     $(\tilde{\boldsymbol{r}}, \beta) \leftarrow$ last item in *solution path*.
7:     $\tilde{r}(\hat{x}) \leftarrow 0$ for any $\hat{x}$ with $\tilde{r}(\hat{x}) < \delta$.
8:     $\tilde{\boldsymbol{r}} \leftarrow$ normalize $\tilde{\boldsymbol{r}}$.
9:     $\tilde{\boldsymbol{r}} \leftarrow$ BLAHUT-ARIMOTO$(\tilde{\boldsymbol{r}}, \beta_n)$.   ▷ Iterate BA_β (2.20) until convergence.
10: **end while**
11: **return** *results*   ▷ Approximations of the true solutions $\boldsymbol{r}(\beta_n)$.

In practice, this heuristic works nicely so long that the bifurcation is not missed altogether due to large approximation errors in Algorithm 2. Off-grid points are later extrapolated using a Taylor step (3.1) at the last grid point, unless the extrapolation has a negative coordinate due to an

---
[9]Note that the step size $\Delta\beta$ is negative.

over-shot bifurcation. In this case, the point is extrapolated using the expansion at the next grid point. As an alternative heuristic, the bifurcation could be handled by extrapolating it linearly from the last point before the threshold is crossed and then using Blahut-Arimoto, exploiting the accuracy of first-order derivatives; cf., Figure 2.3. This also works nicely in practice, on grids dense enough.

3.3 Computational costs and cost-to-error tradeoff

We provide bounds on the computational and memory costs of root-tracking for RD (Algorithm 3), and estimate the tradeoff between its error and computational costs. While the cost of implicit derivatives grows rapidly with the order, counter-intuitively, higher orders usually make much better use of the computational cost. Blahut-Arimoto makes little to no use of the computational effort invested at previous grid points and does not yield off-grid information. In contrast, our algorithm leverages the derivatives at a point to improve the accuracy at subsequent ones, allowing also cheap extrapolation of off-grid points. Despite the many improvements possible to root-tracking RD (in Section 3.4 below), the cost of an entire solution curve is already roughly comparable to BA as is, as seen in Figure 1.2 (Section 1). See also Section II.8, on the complexities of root-tracking in general, with and without tensor memorization, and the complexities of RD derivative tensors.

The computational cost of RD root-tracking (Algorithm 3) is comprised of the cost of implicit derivatives and of Blahut-Arimoto near bifurcations. However, the cost due to BA is typically negligible since our heuristic avoids its critical slowing down near bifurcations (see Section 3.2). Memorizing (caching) each computed tensor $D_{\beta^b, \boldsymbol{r}^m}^{b+m}(Id - BA_\beta)[\boldsymbol{r}]$ (at Theorem 4 in Section 2.3) upon its first appearance reduces the computational costs drastically, as shown in Section II.8. Similarly, also for memorizing implicit derivatives when calculating them recursively in Algorithm 1 (Section 2.2). Our implementation also memorizes some of the quantities en route, further reducing costs. For the most part, these can be discarded once done computing at a point, so the memory costs do *not* scale with the grid size. Thus, the computational cost of Algorithm 3 is essentially determined by the implicit derivatives' cost at a point (times the number of grid points). We provide the following (loose) bounds.

Theorem 6 (Complexity bounds for RD root tracking)**.** *For $l \geq 2$, the computational complexity of Algorithm 3 of order l with tensor memorization is bounded by*

$$O\left(N \cdot (M + l)^{(3l+5/2)} e^{(2M+l)H_e(\rho) + \pi\sqrt{2l/3}}\right) \tag{3.3}$$

times the number of grid-points. Where, N and M are the source and reproduction alphabet sizes (respectively), $\rho := \frac{2M}{2M+l}$, and $H_e(\cdot)$ is the binary entropy in nats. For $l = 1$, it is instead

$$O\left(M^3 N\right) . \tag{3.4}$$

The algorithm's memory complexity is bounded by

$$O\left(l^{(l+5/2)} \ln l\right) + O\left(MNl^2\right) + O\left(M^l l\right) \tag{3.5}$$

The bounds provided by Theorem 6 are rather loose in the derivative's order l. The term $(M + l)^{(3l+5/2)}$ in the computational complexity (3.3) is hyper-exponential in l primarily due to inefficient copy operations used to simplify our implementation, and due to the loose bounds used in the theorem's proof (in Section III.E.3). Similarly, the first term in the memory complexity (3.5) is in practice only of order $O(l \cdot 1.73^l)$, at least for $l \leq 25$; see Table 2 in Section II.8. Nevertheless, the complexity of higher implicit multivariate derivatives is high even when the cost

of RD derivative tensors is set aside (Proposition 30 in Section II.8). This arguably stems from[10] the combinatorics of partial derivatives, [Hardy, 2006]. With that, we note that the complexities of Algorithm 3 (both in practice and the bounds (3.3)-(3.5)) do *not* depend on the problem's details[11], but only on its dimensions N and M, and on the order l; denote by cost-per-grid-point(N, M, l) its computational cost at a grid-point. On the other hand, the algorithm's accuracy does depend on the problem's details.

Although the algorithm's complexities grow rapidly with the order l, higher orders generally provide a much better cost-to-error tradeoff, as shown below. Thanks to our choice of cluster-marginal coordinates (in Section 2.3), the complexities are only linear in the source alphabet size N. This is useful when computing with large source alphabets, $|M| \ll |N|$. For any fixed order l, $e^{(2M+l)H_e(\rho)}$ is asymptotically linear in M, and so the costs are polynomial in M. Further, the reproduction alphabet size M in these bounds can be replaced by the solution's support size, which varies along the grid. For, Algorithm 3 reduces the problem after each bifurcation (in step 3).
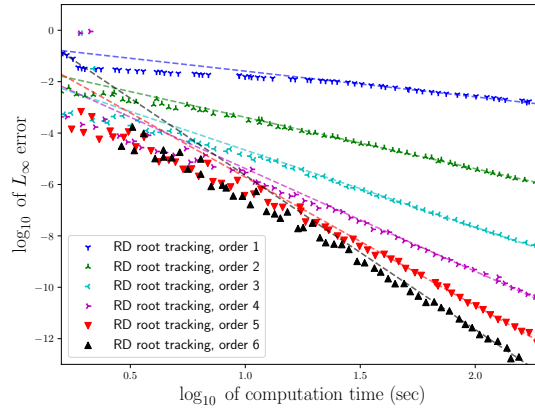

Figure 3.4: **Higher order methods make better use of computational costs when small error is required.** To demonstrate Equation (3.6), a linear regression of slope $(-l)$ (dashed line) is matched to the rightmost markers per method order l. Plot details are as in Figure 1.2.

Ideally, it would be possible to tell the tradeoff between accuracy and computational cost in advance, with better results expected as the computational effort is increased. Indeed, except for δ-close to a bifurcation, an l-th order RD root-tracking with step-size $|\Delta\beta|$ converges to the true solution at a rate of $O(|\Delta\beta|^l)$, by Theorem 5 (in Section 3.1). So, write $error \propto |\Delta\beta|^l$, when the step-size $|\Delta\beta|$ is small. As the number of grid points is inversely proportional to the step size, the total computational cost is of order $|\Delta\beta|^{-1} \cdot$ cost-per-grid-point(N, M, l). This yields

$$\log error \propto l \cdot \log \left(\text{cost-per-grid-point}(N, M, l) \right) - l \cdot \log(\text{total-cost}) . \tag{3.6}$$

Carefully note that, with other parameters fixed, increasing the computational effort reduces the error at a rate of $(\text{total-cost})^{-l}$. Thus, despite their costs, derivatives of higher orders l generally provide a much better tradeoff between the error and computational costs. This off-hand tradeoff is demonstrated by Figure 3.4; it is increasingly accurate at higher costs (smaller $|\Delta\beta|$), as expected. Similar plots are typical to the error analysis of the Taylor method; e.g., [Butcher, 2016, Figure 223(i)].

Unlike Algorithm 3, Blahut-Arimoto makes little use (if any) of the computational cost invested in adjacent grid points. For, BA is usually computed independently along a grid (computing each

---

[10]cf., Theorem 7 (Section II.4), which is used to prove the formula of Theorem 3 (Section 2.2) for higher implicit multivariate derivatives.

[11]e.g., examine the dependence of Equations (2.28)-(2.32) (in Section 2.3) on the problem definition (d, p_X).

point anew) or in reverse annealing Rose et al. [1990], where the solution at one point is taken as the initialization for the next. While reverse annealing reduces the computational costs noticeably, it may follow sub-optimal solution branches. In a sense, reverse annealing is similar to root tracking of zeroth order.

Since Algorithm 3 leverages each point to calculate the next, executing it on denser grids or at higher orders will usually improve the overall accuracy (though not always – see Figure 1.2). In contrast, adding grid points to BA usually degrades its overall accuracy due to critical slowing down, [Agmon et al., 2021]; cf., Figure 2.3 (bottom). This is demonstrated by Figure 1.2. Judging by our implementations of both, it is easy to find parameters where either algorithm outperforms the other. With that, various improvements can be made to Algorithm 3, as we discuss next.

3.4 Possible improvements to RD root-tracking

We discuss several approaches which we believe could significantly improve the cost-to-error tradeoff of RD root tracking (Algorithm 3). Most notably, by making the derivative order and step size adaptive. cf., the related Sections 3.3 and II.7 on the cost-to-error tradeoff and error analysis.

Tracking an operator's root belong in general to a family of hard-to-solve numerical problems called *stiff* (see Section II.7.2). While we acknowledge that much literature exists on stiff differential equations, we have chosen to ignore it to avoid straying off our main line of discussion. Instead, we focus on conspicuous improvements, which this work has not attended for the sake of simplicity.

Algorithm 3 uses both the classic Taylor method for solving ordinary differential equations and Blahut-Arimoto's algorithm. Numerical methods for solving $\frac{d\boldsymbol{x}}{d\beta} = f(\boldsymbol{x}, \beta)$ usually exploit the values of f or its derivatives to approximate $\boldsymbol{x}(\beta)$, but often cannot directly reduce the error accumulated from the true solution $\boldsymbol{x}(\beta_0)$ at a point β_0. cf., Section 22 in Butcher [2016] for example. On the other hand, Blahut-Arimoto's algorithm reduces the error from the true solution, but does not follow its path $\boldsymbol{x}(\beta)$ as β varies. We note that these building blocks could have been combined in ways other than those we have chosen. However, to our understanding, it is necessary to use a component that directly reduces the error accumulated at a point, whether BA or another algorithm. Otherwise, either the error accumulated until a bifurcation would get out of control, or the computational costs. For, following an operator's root is inherently stiff; see Section II.7.2.

Rather than using implicit derivatives to construct the Taylor expansion around a point, they could be used to find the rational fraction which best matches the expansion, or *Padé approximant*; e.g., the classic Baker Jr [1975]. These are often superior to Taylor series, Weisstein.

However, even with the Taylor series method, neither fixing its step size nor its order are optimal choices. For, there may be large spans along the β-axis where the solution changes very little. For example, in Figure 3.2, most of the computational power is spent where the solution barely changes (till point A there). Instead, big step sizes or low orders could have been more cost-effective there, while using small step sizes or high orders where the solution changes rapidly (to the left of A there). On the other hand, numerical derivatives of higher orders lose their accuracy faster when approaching a bifurcation (Figure 2.3 top). An estimate of the local error is needed to make the method adaptive, [Butcher, 2016, 226]. This can be achieved by estimating the local Lipschitz constants of the Taylor expansion (3.1), by computing $D_{\boldsymbol{r}}T_l$. These can be computed easily using only the results presented so far and Proposition 27 (in Section II.7.3).

Last, our implementation for computing the derivative tensors could be optimized. In addition to many non-algorithmic optimizations, note that it copies out each partial derivative to multiple memory locations, which correspond to permutations of the derivatives' order, as elaborated in Section III.E.2.3.

Part II
The ideas underlying root-tracking for rate-distortion

This part elaborates on the main ideas supporting the results of Part I.

Section 4 elaborates on the mathematical prerequisites of arbitrary-order derivative calculations. These are necessary for the formula of Theorem 3 for implicit derivatives of arbitrary-order (Section I.2.2), and for Blahut-Arimoto's derivative tensors (Section I.2.3). Section 5 outlines the proof of the latter, provides related results such as the RD ODE, and comments how to compute RD derivative tensors efficiently.

To reconstruct an RD solution curve from its implicit derivatives, Section 6 provides some understanding of RD bifurcations. While its goal to show that Algorithms 2 and 3 follow the optimal solution path subject to Assumption 3, we provide several basic results on RD bifurcations, and the tools to detect and distinguish between two types of bifurcations.

In Section 7 we analyze the error accumulated by Algorithm 2 (in Section I.3.1). The analysis not only reveals the source of the computational difficulty, but also provides a tool that could be used to mitigate it (see Section I.3.4 on improvements). Finally, Section 8 provides results supporting the computational and memory complexity bounds presented at Section I.3.3: for root tracking in general, for RD root tracking (Algorithm 3), and for computing RD derivative tensors.

4   Preliminaries: the multivariate Faà di Bruno's formula

We review the preliminaries needed to calculate the higher derivative tensors of $Id - BA_\beta$ (1.2) (in Section 5), and for the proof of the arbitrary-order expansion of $\frac{d^l}{d\beta^l}F$ of Theorem 1 (in Section III.B.1). These are based chiefly on the multivariate Faà di Bruno's formula from Ma [2009], in Theorem 7 below.

The n-th derivative of a product $g(t) \cdot f(t)$ of (scalar) functions f and g is given by the well-known Leibniz rule,

$$\frac{d^n}{dt^n}(gf) = \sum_{k=0}^{n} \binom{n}{k}\frac{d^k g}{dt^k} \cdot \frac{d^{n-k}f}{dt^{n-k}} \ . \tag{4.1}$$

In comparison, the n-th derivative of the composite $g(f(t))$ is given by the less familiar Faà di Bruno's formula, Faà di Bruno [1855, 1857]. If all the necessary derivatives of $g(t)$ and $f(t)$ are defined, it can be written [Roman, 1980, Theorem 2] as

$$\frac{d^n}{dt^n}g(f(t)) = \sum \frac{n!}{k_1! \ 1!^{k_1} \cdots k_n! \ n!^{k_n}} \cdot g^{(k_1+\cdots+k_n)}(f(t)) \cdot \left(f^{(1)}(t)\right)^{k_1} \cdots \left(f^{(n)}(t)\right)^{k_n}, \tag{4.2}$$

where the sum is over all the non-negatives integers $k_1, \ldots, k_n$ satisfying $k_1 + 2k_2 + \cdots + nk_n = n$. These represent a *partition* (as in Section I.2.2) of an n-sized set to k_1 subsets of size 1, k_2 subsets of size 2, etc., hence $k_1 + k_2 + \cdots + k_n$ subsets in total. cf., Hardy [2006] for a lucid combinatorial interpretation of the formula's coefficients, Johnson [2002] for further details and historical notes.

For our purposes, we shall also need a multivariate version of Faà di Bruno's formula (4.2). While its multivariate form is often attributed to[12] Constantine and Savits [1996, Theorem 2.1], we present a more modern form by Ma [2009], which generalizes the combinatorial arguments of Hardy [2006] from the univariate case.

---

[12]cf., literature survey at the introduction of Ma [2009].

Following Ma [2009], we first recall some multivariate notation. Denote by $\mathbb{N}_0$ the non-negative integers, and let $\boldsymbol{x} \in \mathbb{R}^\nu$. For a *multi-index* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_\nu) \in \mathbb{N}_0^\nu$ of *length* ν, define

$$|\boldsymbol{\alpha}| := \sum_{j=1}^{\nu} \alpha_j, \quad \boldsymbol{\alpha}! := \prod_{j=1}^{\nu} \alpha_j!, \quad \boldsymbol{x}^{\boldsymbol{\alpha}} := \prod_{j=1}^{\nu} x_j^{\alpha_j}, \quad \frac{\partial^{|\boldsymbol{\alpha}|} z}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} := \prod_{j=1}^{\nu} \left(\frac{\partial}{\partial x_j} \right)^{\alpha_j} z, \tag{4.3}$$

with θ^0 defined to be 1. A multi-index $\boldsymbol{\alpha} \in \mathbb{N}_0^\nu$ is said to be *decomposed* into s *parts* $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_s \in \mathbb{N}_0^\nu$ with (respective) *multiplicities* $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_s \in \mathbb{N}_0^\mu$ if the *decomposition equation*

$$\boldsymbol{\alpha} = |\boldsymbol{m}_1| \boldsymbol{p}_1 + |\boldsymbol{m}_2| \boldsymbol{p}_2 + \cdots + |\boldsymbol{m}_s| \boldsymbol{p}_s \tag{4.4}$$

holds, and the parts are distinct. Note that the parts $\boldsymbol{p}$'s and the multiplicities $\boldsymbol{m}$'s are multi-indices of lengths ν and μ, respectively. The *total multiplicity* is defined to be

$$\boldsymbol{m} := \boldsymbol{m}_1 + \boldsymbol{m}_2 + \cdots + \boldsymbol{m}_s. \tag{4.5}$$

The list $(s, \boldsymbol{p}, \boldsymbol{m})$ is a μ-*decomposition* of α, or simply a decomposition. One way to ensure that its parts are distinct is by requiring $0 \ll \boldsymbol{p}_1 \ll \boldsymbol{p}_2 \ll \cdots \ll \boldsymbol{p}_s$, Ma [2009], where $\ll$ is the total order defined by $\boldsymbol{\alpha} \ll \boldsymbol{\beta}$ if there is $j \leq \nu$ such that $\alpha_1 = \beta_1, \ldots, \alpha_{j-1} = \beta_{j-1}$, but $\alpha_j < \beta_j$.

Theorem 7 (Multivariate Faà di Bruno's, Ma [2009]). *Let $\boldsymbol{x} \in \mathbb{R}^\nu \xrightarrow{f} \boldsymbol{y} \in \mathbb{R}^\mu \xrightarrow{g} z \in \mathbb{R}$, with f, g sufficiently smooth functions. Write $\boldsymbol{x} = (x_1, \ldots, x_\nu)$ and $\boldsymbol{y} = (y_1, \ldots, y_\mu)$ for the variables' coordinates. Then,*

$$\frac{\partial^{|\boldsymbol{\alpha}|} z}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \boldsymbol{\alpha}! \sum_{(s, \boldsymbol{p}, \boldsymbol{m}) \in \mathcal{D}} \frac{\partial^{|\boldsymbol{m}|} z}{\partial \boldsymbol{y}^{\boldsymbol{m}}} \prod_{k=1}^{s} \frac{1}{\boldsymbol{m}_k!} \left[\frac{1}{\boldsymbol{p}_k!} \frac{\partial^{|\boldsymbol{p}_k|} \boldsymbol{y}}{\partial \boldsymbol{x}^{\boldsymbol{p}_k}} \right]^{\boldsymbol{m}_k} \tag{4.6}$$

where $\mathcal{D}$ is the set of all μ-decompositions of $\boldsymbol{\alpha}$. Explicitly, the summands to the right are given by

$$\boldsymbol{\alpha}! \frac{\partial^{r_1 + \cdots + r_\mu} z}{\partial y_1^{r_1} \cdots \partial y_\mu^{r_\mu}} \prod_{k=1}^{s} \prod_{i=1}^{\mu} \frac{1}{m_{k,i}!} \left[\frac{1}{p_{k,1}! \cdots p_{k,\nu}!} \frac{\partial^{p_{k,1} + \cdots + p_{k,\nu}} y_i}{\partial x_1^{p_{k,1}} \cdots \partial x_\nu^{p_{k,\nu}}} \right]^{m_{k,i}} \tag{4.7}$$

where $\boldsymbol{p}_k = (p_{k,1}, \cdots, p_{k,\nu}), \boldsymbol{m}_k = (m_{k,1}, \cdots, m_{k,\mu}), r_i := m_{1,i} + \cdots + m_{s,i}$, and hence $\boldsymbol{m} = (r_1, \cdots, r_\mu)$.

The multivariate Faà di Bruno's formula Ma [2009] at (4.6) simplifies when the outer composite g in $g \circ f$ is linear. The first term $\frac{\partial^{|\boldsymbol{m}|} z}{\partial \boldsymbol{y}^{\boldsymbol{m}}}$ at (4.6) then vanishes, except for decompositions of a total multiplicity $|\boldsymbol{m}| = 1$. From definition (4.5), there is only one part of non-zero multiplicity in such decompositions. They may be taken with $s = 1$, $\boldsymbol{p}_1 = \boldsymbol{\alpha}$ and total multiplicity $\boldsymbol{m} = \boldsymbol{m}_1$ (compare to (4.6) and the decomposition equation (4.4)). That is, such μ-decompositions are determined by picking a unit vector $\boldsymbol{m}_1 = \boldsymbol{e}_j \in \mathbb{N}_0^\mu$, for $j = 1, \ldots, \mu$. We obtain,

Corollary 8. *Under the conditions of Theorem 7, suppose that $g : \mathbb{R}^\mu \to \mathbb{R}$ is linear. Then,*

$$\frac{\partial^{|\boldsymbol{\alpha}|} z}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} = \sum_{j=1}^{\mu} \frac{\partial z}{\partial y_j} \cdot \frac{\partial^{|\boldsymbol{\alpha}|} y_j}{\partial \boldsymbol{x}^{\boldsymbol{\alpha}}} \tag{4.8}$$

5 High order derivatives of the Blahut-Arimoto operator in cluster-marginal coordinates

In this section we outline the proof of the closed-form formulae for the higher derivative tensors of $Id - BA_\beta$ (1.2) at Theorem 4 (in Section I.2.3). See Section III.C for details. On top of intermediate results, Theorem 14 below (Section 5.3) provides the specialization to RD of the implicit differential equation $D_{\boldsymbol{x}} F \frac{d\boldsymbol{x}}{d\beta} = -D_\beta F$ (2.4). The efficient computations of the derivatives tensors of $Id - BA_\beta$ is discussed in Section 5.5. The results here are built on top of Section 4, primarily on the multivariate Faà di Bruno's formula at Ma [2009] (Theorem 7).

Recall the notations of Section I.2.3. Let p_X and $d(x,\hat{x})$ define an RD problem. Write $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$ for the source and reproduction alphabet sizes, respectively. The BA_β operator in marginal coordinates (2.20) is defined there as the composition of

$$q(\hat{x}|x) := \frac{r(\hat{x})\,e^{-\beta d(x,\hat{x})}}{Z(x,\beta)} \quad \text{and} \tag{2.18}$$

$$s(\hat{x}) := \sum_x p_X(x) q(\hat{x}|x)\,, \tag{2.19}$$

in that order, where $Z(x,\beta) := \sum_{\hat{x}'} r(\hat{x}') e^{-\beta d(x,\hat{x}')}$. We write $\boldsymbol{s}$ rather than $\boldsymbol{r}$ at the cluster-marginal equation (2.19) to better distinguish input from output marginals. While our eventual goal is to track RD solutions, calculations in this section do *not* assume $\boldsymbol{r}$ to be a fixed point of BA_β, $\boldsymbol{s} := BA_\beta[\boldsymbol{r}] = \boldsymbol{r}$, unless stated otherwise.

The encoder equation (2.18) can be considered as $M \times N$ real functions $q(\hat{x}|x)$ in the $1 + M$ variables β and $r(\hat{x})$, while the cluster marginal equation (2.19) are M functions $s(\hat{x})$ in the $M \times N$ variables $q(\hat{x}|x)$. That is, we view a Blahut-Arimoto iteration (2.20) as the composition

$$(\beta, \boldsymbol{r}) \in \mathbb{R}^{1+M} \xrightarrow{(2.18)} \boldsymbol{q} \in \mathbb{R}^{M \times N} \xrightarrow{(2.19)} \boldsymbol{s} \in \mathbb{R}^M\,. \tag{5.1}$$

Of these two steps, the marginal equation (2.19) is linear. That renders the derivatives of $\boldsymbol{s}$ with respect to $\boldsymbol{q}$ rather straightforward, in Section 5.1. As the encoder equation (2.18) is more complicated, we untangle its high-order derivatives gradually. In Section 5.2 we tackle the repeated derivatives of $\boldsymbol{q}$ with respect to $\boldsymbol{r}$ alone, and in Section 5.3 with respect to β alone. Only in 5.4 do we combine derivatives of both types, yielding formula (2.31) for mixed derivatives of arbitrary order (Theorem 4 in Section I.2.3).

5.1 Deriving the marginal equation (2.19)

Denote high-order derivatives using multi-index notation, as in the multivariate Faà di Bruno's formula (in Section 4). A derivative of BA_β (2.20) with respect to its $1 + M$ variables $(\beta, \boldsymbol{r})$ is denoted by $\boldsymbol{\alpha} \in \mathbb{N}_0^{1+M}$; cf., the dependencies depicted in Equation (5.1). Considering β as the zeroth coordinate, α_0 stands for the number of differentiations with respect to β, and α_j for that with respect to $r(\hat{x}_j)$, where $1 \le j \le M$. We write $\boldsymbol{\alpha}_+$ for $\boldsymbol{\alpha}$ with its zeroth coordinate removed, $(\alpha_0, \boldsymbol{\alpha}_+) := (\alpha_0, \alpha_1, \ldots, \alpha_M)$.

Next, exploiting the linearity of the marginal's equation (2.19), we apply Corollary 8 to the multivariate Faà di Bruno's formula (of Section 4). As the $\hat{x}'$ output coordinate of BA_β is a real function, we have

$$\frac{\partial^{|\boldsymbol{\alpha}|} BA_\beta\,[\boldsymbol{r}]\,(\hat{x}')}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} \stackrel{(4.8)}{=} \sum_{x,\hat{x}} \frac{\partial s(\hat{x}')}{\partial q(\hat{x}|x)} \frac{\partial^{|\boldsymbol{\alpha}|} q(\hat{x}|x)}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}}\,. \tag{5.2}$$

For the marginal equation's (2.19) first order derivative,

$$\frac{\partial}{\partial q(\hat{x}|x)} s(\hat{x}') = \sum_{x'} p_X(x') \frac{\partial}{\partial q(\hat{x}|x)} q(\hat{x}'|x') = \delta_{\hat{x},\hat{x}'} \cdot p_X(x) \tag{5.3}$$

As expected, this is constant in the variables $q(\hat{x}|x)$, and so all the higher derivatives of the output marginal in (2.19) vanish. Thus,

$$\frac{\partial^{|\boldsymbol{\alpha}|} BA_\beta [\boldsymbol{r}](\hat{x}')}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} \overset{(5.2)}{=} \sum_{x,\hat{x}} \frac{\partial s(\hat{x}')}{\partial q(\hat{x}|x)} \frac{\partial^{|\boldsymbol{\alpha}|} q(\hat{x}|x)}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}}$$

$$\overset{(5.3)}{=} \sum_{x,\hat{x}} \delta_{\hat{x},\hat{x}'} \cdot p_X(x) \frac{\partial^{|\boldsymbol{\alpha}|} q(\hat{x}|x)}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} = \sum_x p_X(x) \frac{\partial^{|\boldsymbol{\alpha}|} q(\hat{x}'|x)}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} \tag{5.4}$$

To complete the calculation, we calculate the encoder's derivatives in the following subsections.

5.2 Encoder's (2.18) derivatives with respect to the marginal

Repeatedly deriving the encoder (2.18) with respect to the coordinates of the input marginal $\boldsymbol{r}$ yields the following; it is proved in Section III.C.1 by induction on the order $|\boldsymbol{\alpha}_+|$ of differentiation.

Proposition 9. *Let $\beta > 0$, $\boldsymbol{r} \in \mathbb{R}^M$ a distribution, and $\boldsymbol{\alpha}$ a multi-index as above. Let $\boldsymbol{q}$ be defined in terms of $\boldsymbol{r}$ by the encoder Equation (2.18). Its derivative of order $\boldsymbol{0} \neq \boldsymbol{\alpha}_+ \in \mathbb{N}_0^M$ with respect to $\boldsymbol{r}$ is,*

$$\frac{\partial^{|\boldsymbol{\alpha}_+|}}{\partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} q(\hat{x}'|x) \Big|_{\boldsymbol{r}} = \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+| - 1)! \; e^{-\beta \langle \boldsymbol{\alpha}_+, d(x,\hat{x}) \rangle}}{Z^{|\boldsymbol{\alpha}_+|}(x,\beta)} \cdot \Big[\langle \boldsymbol{\alpha}_+, \boldsymbol{e}_{\hat{x}'} \rangle - |\boldsymbol{\alpha}_+| \cdot q(\hat{x}'|x) \Big] \tag{5.5}$$

where $\langle \cdot, \cdot \rangle$ is the usual scalar product on $\mathbb{R}^M$, $\boldsymbol{e}_{\hat{x}'}$ is the standard basis vector in $\mathbb{R}^M$ at the $\hat{x}'$ entry, and $d(x,\hat{x})$ is considered as an $\hat{x}$-indexed vector for x fixed.

To clarify (5.5), when $x \in \mathcal{X}$ is fixed, then $d(x,\hat{x})$ is merely a vector in $\mathbb{R}^M$. As $\boldsymbol{\alpha}_+$ is in $\mathbb{N}_0^M \subset \mathbb{R}_0^M$, there is sense in taking the scalar product $\langle \boldsymbol{\alpha}_+, d(x,\hat{x}) \rangle$. For example, when $\boldsymbol{\alpha}_+$ is the j-th standard basis vector $\boldsymbol{e}_j$, then

$$\langle \boldsymbol{e}_j, d(x,\hat{x}) \rangle = d(x,\hat{x}_j) . \tag{5.6}$$

The first-order derivatives $|\boldsymbol{\alpha}_+| = 1$ are an important special case,

Corollary 10. *Outside the simplex boundary, $\forall \hat{x} \; r(\hat{x}) > 0$, the Jacobian of $Id - BA_\beta$ (1.2) in marginal coordinates when evaluated at $\boldsymbol{r}$ is given by*

$$\left(D_{\boldsymbol{r}} \left(Id - BA_\beta \right) \big|_{\boldsymbol{r}} \right)_{\hat{x}_i,\hat{x}_j} = \sum_x p_X(x) \frac{q(\hat{x}_j|x)q(\hat{x}_i|x)}{r(\hat{x}_j)} + \Delta[\boldsymbol{r}]_{\hat{x}_i,\hat{x}_j} \tag{5.7}$$

where $\boldsymbol{q}$ is defined in terms of $\boldsymbol{r}$ by the encoder Equation (2.18), and $\Delta[\boldsymbol{r}] := \mathrm{diag} \left(\frac{r(\hat{x}_j) - BA_\beta[\boldsymbol{r}](\hat{x}_j)}{r(\hat{x}_j)} \right)_j$.

The term $\Delta[\boldsymbol{r}]$ vanishes precisely at fixed points of BA_β, $\boldsymbol{r} = BA_\beta[\boldsymbol{r}]$. It can be considered as a perturbation due to evaluating the Jacobian outside of fixed points. We shall need the general form (5.7) for the proof of the convergence guarantees Theorem 5 of Taylor method for RD. Otherwise, when evaluated at fixed points, this result agrees with Agmon et al. [2021] (A^{T} at Equation (7) there). While the Jacobian can also be calculated at the simplex boundary, that shall not be useful to us. Rather than calculating (5.7) directly, we prove it using Proposition 9 to illustrate the multivariate notation.

Proof of Corollary 10. A first order derivative $\partial/\partial r(\hat{x}_j)$ is represented by the j-th standard basis vector, $\boldsymbol{\alpha}_+ = \boldsymbol{e}_j$. Differentiating its $(\hat{x}_i, x)$ entry,

$$\frac{\partial^{|\boldsymbol{\alpha}_+|}}{\partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} q(\hat{x}_i|x) = \frac{\partial}{\partial r(\hat{x}_j)} q(\hat{x}_i|x) \overset{(5.5)}{\underset{(5.6)}{=}} \frac{e^{-\beta d(x,\hat{x}_j)}}{Z(x,\beta)} \cdot \left[\langle \boldsymbol{e}_j, \boldsymbol{e}_i \rangle - q(\hat{x}_i|x) \right] \overset{(2.18)}{=} \frac{q(\hat{x}_j|x)}{r(\hat{x}_j)} \left[\delta_{i,j} - q(\hat{x}_i|x) \right] \tag{5.8}$$

Setting $\alpha_0 = 0$ and plugging this back into formula (5.4) for the derivative of BA_β yields (5.7),

$$\frac{\partial BA_\beta \left[\boldsymbol{r} \right] (\hat{x}_i)}{\partial r(\hat{x}_j)} \overset{(5.4)}{=} \sum_x p_X(x) \frac{\partial q(\hat{x}_i|x)}{\partial r(\hat{x}_j)} \overset{(5.8)}{=} \sum_x p_X(x) \frac{q(\hat{x}_j|x)}{r(\hat{x}_j)} \left[\delta_{i,j} - q(\hat{x}_i|x) \right]$$

$$\overset{(2.19)}{=} \delta_{i,j} \cdot \frac{BA_\beta[\boldsymbol{r}](\hat{x}_j)}{r(\hat{x}_j)} - \sum_x p_X(x) \frac{q(\hat{x}_j|x)q(\hat{x}_i|x)}{r(\hat{x}_j)} \tag{5.9}$$

Where at the last equality, $BA_\beta[\boldsymbol{r}](\hat{x}_j)$ is the j-th output coordinate $s(\hat{x}_j)$ (2.19) of a Blahut-Arimoto iteration calculated at $\boldsymbol{r}$. $\qquad\square$

At a fixed point of BA_β, the Jacobian's properties relevant to us are the following, due to Benger.

Theorem 11 (Agmon et al. [2021]). *Let an RD problem be defined by p_X and a finite non-degenerate distortion $d(x,\hat{x})$ (defined in Section I.2.3). Let $\boldsymbol{r}$ be a fixed point of BA_β outside the simplex boundary, $\forall \hat{x}\ r(\hat{x}) > 0$. Then, $D_{\boldsymbol{r}} \left(Id - BA_\beta \right)|_{\boldsymbol{r}}$ (5.7) is non-singular, diagonalizable, and with real non-negative eigenvalues.*

5.3   Encoder's (2.18) partial derivative with respect to β

For derivatives with respect to β we need the polynomials P_k (2.23)-(2.24) defined in Section I.2.3. Recall the derivation $\bar{d}$ defined there on the infinite polynomial ring $\mathbb{R}\left[x_0, x_1, \dots \right]$,

$$\bar{d}x_0 := 0, \quad \text{and} \quad \bar{d}x_k := x_1 \cdot x_k - x_{k+1} \quad \text{for } k > 0 . \tag{2.22}$$

To calculate $\bar{d}$ on a monomial, apply the Leibniz rule $\bar{d}x_i^j = jx_i^{j-1}\bar{d}x_i$, for $j > 0$,

$$\bar{d}\left(x_0^{i_0} x_1^{i_1} x_2^{i_2} \cdots x_k^{i_k} \right)$$
$$= \left(i_0 x_0^{i_0-1} \bar{d}x_0 \right) x_1^{i_1} x_2^{i_2} \cdots x_k^{i_k} + x_0^{i_0} \left(i_1 x_1^{i_1-1} \bar{d}x_1 \right) x_2^{i_2} \cdots x_k^{i_k} + \cdots + x_0^{i_0} x_1^{i_1} x_2^{i_2} \cdots \left(i_k x_k^{i_k-1} \bar{d}x_k \right)$$
$$\overset{(2.22)}{=} 0 + i_1 x_0^{i_0} x_1^{i_1-1} \left(x_1^2 - x_2 \right) x_2^{i_2} \cdots x_k^{i_k} + \cdots + i_k x_0^{i_0} x_1^{i_1} x_2^{i_2} \cdots x_k^{i_k-1} \left(x_1 x_k - x_{k+1} \right) \tag{5.10}$$

where $i_0, i_1, \dots, i_k \geq 0$, and the j-th summand is understood to vanish if $i_j = 0$. The first equality follows from the usual rules of differentiation, the second from the definition (2.22) of $\bar{d}$. With this, one can calculate $\bar{d}P_k$ for a polynomial P_k, by linearity of $\bar{d}$. Starting at $P_0(x_0) = 1$ (2.23), use the inductive definition (2.24) to calculate P_{k+1} from P_k. e.g., P_1, P_2 and P_3 are given at (2.25)-(2.27).

The polynomials P_k are defined by simple algebraic formulae (2.23)-(2.24) which can be calculated easily, as in (5.10). They encapsulate the algebra involved in the encoder's high-order partial derivatives with respect to β. Plugging the distortion $d(x,\hat{x})$ and the expectations $\mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right]$ (2.28) in place of the variables $x_0, x_1, \dots, x_k$ of P_k, one can calculate the M-by-N matrices $P_k[\boldsymbol{q}; d](\hat{x}, x)$ (2.29), also denoted $P_k(\hat{x}, x)$ for short. This yields the following formula for the encoder's repeated partial β-derivatives — see Section III.C.2 for proof.

Proposition 12. *Let $\beta > 0$, $\boldsymbol{r} \in \mathbb{R}^M$ a distribution, and $\boldsymbol{q}$ defined in terms of $\boldsymbol{r}$ by the encoder Equation (2.18). Then for $k > 0$,*

$$\left. \frac{\partial^k q(\hat{x}|x)}{\partial \beta^k} \right|_{\boldsymbol{r}} = q(\hat{x}|x) \cdot P_k(\hat{x}, x) \tag{5.11}$$

Plugging this into the expansion (5.4) of $\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial\beta^{\alpha_0}\partial\boldsymbol{r}^{\boldsymbol{\alpha}+}}BA_\beta\left[\boldsymbol{r}\right](\hat{x}')$ (in Section 5.1) immediately yields formula (2.30) of Theorem 4 (Section I.2.3) for the repeated β-derivative of $Id - BA_\beta$ (1.2). With P_1 written explicitly, the first order can be written explicitly as follows (see Section III.C.3 for proof),

Corollary 13. *With the notation of Proposition 12,*

$$\frac{\partial}{\partial\beta}\left(Id - BA_\beta\right)\Big|_{\boldsymbol{r}}(\hat{x}) = \mathbb{E}_{p_X(x)}\left[q(\hat{x}|x)d(x,\hat{x})\right] - \mathbb{E}_{q(\hat{x}'|x)p_X(x)}\left[q(\hat{x}|x)d(x,\hat{x}')\right] \tag{5.12}$$

Now that Corollaries 10 and 13 provide all the first-order derivative tensors, we can specialize the implicit differential equation $D_{\boldsymbol{x}}F\,\frac{d\boldsymbol{x}}{d\beta} = -D_\beta F$ (2.4) to RD.

Theorem 14 (RD ODE). *Let $\boldsymbol{r}$ be a fixed point of BA_β outside the simplex boundary, $\forall\hat{x}\ r(\hat{x}) > 0$, for which Assumptions 1 and 2 in Section I.2.1 hold. Then*

$$\sum_{\hat{x}'}A_{\hat{x},\hat{x}'}\frac{dr(\hat{x}')}{d\beta} = \mathbb{E}_{q(\hat{x}'|x)p_X(x)}\left[q(\hat{x}|x)d(x,\hat{x}')\right] - \mathbb{E}_{p_X(x)}\left[q(\hat{x}|x)d(x,\hat{x})\right] \tag{5.13}$$

where $\boldsymbol{q}$ is defined in terms of $\boldsymbol{r}$ by the encoder Equation (2.18), and

$$A_{\hat{x},\hat{x}'} := \sum_x p_X(x)\frac{q(\hat{x}'|x)q(\hat{x}|x)}{r(\hat{x}')}\ . \tag{5.14}$$

5.4 Encoder's (2.18) mixed derivatives

The previous Sections 5.2 and 5.3 enabled us to calculate derivatives with respect to the coordinates $\boldsymbol{r}$ alone or with respect to β alone. cf., the dependencies Equation (5.1). To derive simultaneously with respect to both, recall $G(k,a;\boldsymbol{q},d)$ (2.32) from Section I.2.3. It is defined on non-negative integers k,a, and its values are $M\times N$ matrices. We set $G = 0$ if $a = 0 < k$, and otherwise

$$G(k,a;\boldsymbol{q},d)_{(\hat{x},x)} := \sum_{\substack{\boldsymbol{t}:\ |\boldsymbol{t}|\leq a,\\ \sum_j j\cdot t_j = k}} \frac{1}{\boldsymbol{t}!\,(a-|\boldsymbol{t}|)!}\prod_{j=1}^k\left(\frac{P_j(\hat{x},x)}{j!}\right)^{t_j}\ , \tag{2.32}$$

where $\boldsymbol{t}\in\mathbb{N}_0^k$, and $P_j(\hat{x},x)$ are the matrices defined above (Equation (2.29) in Section I.2.3). Since $\boldsymbol{t}$ represents the multiplicities of an integer partition of k, we may take $\boldsymbol{t}\in\mathbb{N}_0^l$ for any convenient $l\geq k$. With that,

Proposition 15. *Let $d(x,\hat{x})$ and p_X define an RD problem, $\boldsymbol{r}\in\Delta[\hat{\mathcal{X}}]$ be an input marginal outside the simplex boundary, $\forall\hat{x}\ r(\hat{x}) > 0$, and $\boldsymbol{q}$ the encoder its defines by the encoder Equation (2.18). For $\boldsymbol{\alpha}\in\mathbb{N}_0^{M+1}$ a multi-index with $\boldsymbol{\alpha}_+\neq\boldsymbol{0}$,*

$$\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial\beta^{\alpha_0}\,\partial\boldsymbol{r}^{\boldsymbol{\alpha}+}}q(\hat{x}'|x)\Big|_{\boldsymbol{r}} = (-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!\left(\frac{q(\hat{x}|x)}{r(\hat{x})}\right)^{\boldsymbol{\alpha}_+}\boldsymbol{\alpha}!\sum_{\boldsymbol{k}:\ |\boldsymbol{k}|=\alpha_0}\left(\prod_{i\neq\hat{x}'}G(k_i,\alpha_i;\boldsymbol{q},d)_{(\hat{x}_i,x)}\right)$$

$$\cdot\left[\alpha_{\hat{x}'}\cdot G(k_{\hat{x}'},\alpha_{\hat{x}'};\boldsymbol{q},d)_{(\hat{x}',x)} - |\boldsymbol{\alpha}_+|\cdot(1+\alpha_{\hat{x}'})\cdot q(\hat{x}'|x)\cdot G(k_{\hat{x}'},1+\alpha_{\hat{x}'};\boldsymbol{q},d)_{(\hat{x}',x)}\right] \tag{5.15}$$

where $\boldsymbol{k}\in\mathbb{N}_0^M$, $i = 1,\ldots,M$, and $G(k,a;\boldsymbol{q},d)$ is defined by (2.32).

This Proposition is proven in Section III.C.4 by differentiating the encoder's $\boldsymbol{r}$-derivative (5.5) with respect to β, followed by an application of formula (5.11) for the encoder's repeated partial β-derivative, using the tools of Section 4. Plugging (5.15) back into the expansion (5.4) of $\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}+}} BA_\beta$, we finally obtain formula (2.31) of Theorem 4 for the mixed derivative of $Id - BA_\beta$ (1.2), stated as Corollary 16 below. Together with Proposition 12 from the previous Subsection, this concludes the proof of Theorem 4 in Section I.2.3.

Corollary 16. *Under the conditions of Proposition 15,*

$$\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}+}} \left(Id - BA_\beta\right) [\boldsymbol{r}] (\hat{x}')$$

$$= \delta_{\boldsymbol{\alpha}, \boldsymbol{e}_{\hat{x}'}} - (-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+| - 1)! \, \boldsymbol{\alpha}! \sum_x p_X(x) \left(\frac{q(\hat{x}|x)}{r(\hat{x})}\right)^{\boldsymbol{\alpha}_+} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \left(\prod_{i \neq \hat{x}'} G\big(k_i, \alpha_i; \boldsymbol{q}, d\big)_{(\hat{x}_i, x)}\right)$$

$$\cdot \left[\alpha_{\hat{x}'} \cdot G\big(k_{\hat{x}'}, \alpha_{\hat{x}'}; \boldsymbol{q}, d\big)_{(\hat{x}', x)} - |\boldsymbol{\alpha}_+| \cdot (1 + \alpha_{\hat{x}'}) \cdot q(\hat{x}'|x) \cdot G\big(k_{\hat{x}'}, 1 + \alpha_{\hat{x}'}; \boldsymbol{q}, d\big)_{(\hat{x}', x)}\right] \quad (5.16)$$

As a sanity check, when differentiating only with respect to the marginal $\boldsymbol{r}$, $\alpha_0 = 0$, then Proposition 15 reduces to formula (5.5) for $\frac{\partial^{|\boldsymbol{\alpha}_+|}}{\partial \boldsymbol{r}^{\boldsymbol{\alpha}+}} q(\hat{x}|x)$, as one might expect. Indeed, $\boldsymbol{k} = \boldsymbol{0}$ is then the only vector in $\mathbb{N}_0^M$ satisfying $|\boldsymbol{k}| = \alpha_0 = 0$. The sum $G(0, a)$ at (2.32) then reduces to $1/a!$, and so the summation at (5.15) over $\boldsymbol{k}$ simplifies to $\frac{1}{\boldsymbol{\alpha}!} \cdot \left[\alpha_{\hat{x}'} - |\boldsymbol{\alpha}_+| \cdot q(\hat{x}'|x)\right]$. Plugging this back into (5.15) yields formula (5.5), as expected. The other special case when differentiating with respect to β alone cannot be verified in a similar manner, since Proposition 15 requires $\boldsymbol{\alpha}_+ \neq 0$.

5.5 A note on how to compute high-order derivatives of $Id - BA_\beta$ efficiently

Some quantities in the formulas for the derivative tensors $D^l_{\beta^b, \boldsymbol{x}^{l-b}}(Id - BA_\beta)[\boldsymbol{r}]$ (Theorem 4 in Section I.2.3) are shared, especially when partial derivatives with respect to β are present. While this was purposefully implied by our presentation, we now discuss how these derivative tensors can computed in an algorithmically efficient manner. cf., Section III.E.2 on the complexities of the derivative tensors.

The form of G (2.32) suggests a multi-step approach to computing the derivatives tensors at Theorem 4. First, compute the M-by-N matrices $P_k(\hat{x}, x)$ (2.29). While the matrices $P_k(\hat{x}, x)$ depend both on $\boldsymbol{q}$ and $d(x, \hat{x})$, their polynomial form P_k depends on k alone, as it is defined algebraically by (2.23)-(2.24). Thus, the P_k polynomials can be computed once an for all, while the $P_k(\hat{x}, x)$ matrices need to be computed anew for every encoder $\boldsymbol{q}$. Yet, the expectations $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')^k\right]$ (2.28) are shared among $P_k(\hat{x}, x)$'s for distinct k values. With these computed, the derivative tensors (2.30) with respect to β alone can be computed readily.

Second, compute the M-by-N matrices $G(k, a)$ according to (2.32), when the $P_k(\hat{x}, x)$ matrices are given. When the range of admissible k and a values is known in advance, then it is possible to iterate over the partition vectors $\boldsymbol{t}$ at (2.32) only once. For, the summand computed there for a particular partition $\boldsymbol{t}$ can be added to the matrices $G(k, a)$ at $k = \sum_j j \cdot t_j$ and at all the a values with $a \geq |\boldsymbol{t}|$.

Third, once the $G(k, a)$ matrices are computed, then one can compute the mixed derivatives of $(Id - BA_\beta)[\boldsymbol{r}]$ by (2.31). Each $\boldsymbol{\alpha}$-indexed partial derivative there may correspond to multiple tensor entries. cf., the comments on indexation, after definition (2.6) in Section I.2.1. Thus, partial derivatives can be computed only once per $\boldsymbol{\alpha}$ and then distributed to the various tensor entries. Alternatively, each $\boldsymbol{\alpha}$-indexed derivative can be re-used at tensor evaluation. See also Section III.E.2.3 on the tensors' complexities.

If one wishes to compute implicit derivatives $\frac{d^l \boldsymbol{r}}{d\beta^l}$ (1.4) of all orders l up to $L > 0$, then inspecting (2.31) shows that it is enough to compute the $P_k(\hat{x}, x)$ matrices for all $k \leq L$, and $G(k, a)$ on the integral grid $0 \leq k \leq L$, $0 \leq a \leq 1 + L$. Once these are computed, then any $\boldsymbol{\alpha}$-indexed partial derivative with $|\boldsymbol{\alpha}| \leq L$ can be computed.

6   On RD bifurcations and root tracking for RD

As described in Section 1, the goal of this work is to track the path of an *optimal* solution. In RD context, this means tracking an *achieving distribution*, rather than any fixed point of the Blahut-Arimoto algorithm. As shown below, there are typically many fixed points of BA_β (2.20) which do *not* achieve the rate-distortion curve, and so are *sub-optimal*. Ensuring that the root being tracked is indeed an optimal one requires an understanding of the solutions' structure or equivalently of RD bifurcations.

We start by showing in Section 6.1 that an RD problem typically has a plethora of suboptimal solutions, stemming from the various *restrictions* of a given problem (defined in Section I.3.1). Our main case of interest is that of a *cluster vanishing*. Namely, when the marginal probability of a cluster $\hat{x}$ vanishes gradually as β varies, resulting in an optimal solution on a smaller support. In Section 6.2 we show that two solution branches must then collide and merge into one, so these are indeed bifurcations. This type of bifurcations is handled by Algorithms 2 and 3 in Section I.3. Besides bifurcations in which the support shrinks, we are also aware of bifurcations where the optimal solution switches support. An explanation of these is deferred to Section 6.5. Both of the latter are *local* bifurcations, as they can be detected by the relevant Jacobian. "There are also bifurcations that cannot be detected by looking at small neighborhoods of fixed points", [Kuznetsov, 2004], known as *global bifurcations*. Such bifurcations could break the continuity of $\boldsymbol{x}(\beta)$, violating Assumption 2. It turns out that there are no global bifurcations in rate-distortion (Section 6.3). However, considering only the cluster-marginal of a root might cause a support-switching bifurcation to appear as if it is a global bifurcation. With that, while we do *not* argue to classify all RD bifurcations, we do believe that the tools brought here are a significant step in that direction.

To track an operator's root $\boldsymbol{x}$ it is necessary that it can be written as a function $\boldsymbol{x}(\beta)$ of β (Assumption 1 in I.2.1) which is smooth (Assumption 2), except at bifurcations. In Section 6.3 we consider the obstructions to these assumptions in RD, building on the classic results of Berger [1971]. Our main result is Theorem 20, which allows easy detection of non-uniqueness of the achieving distribution, in terms of Blahut-Arimoto's Jacobian with respect to the encoder $\boldsymbol{q}$. The subtle differences between the Jacobian in encoder $\boldsymbol{q}$ and in marginal coordinates $\boldsymbol{r}$ allows to detect the bifurcation's type in some cases (Equation (6.6)), and extends the argument of Agmon et al. on critical slowing down. This not only allows us to argue in Section 6.4 that root tracking for RD does indeed track the optimal solution, subject to Assumption 3, but also gives us a straightforward tool to detect failures, and so to consider how they might be corrected.

We note that Rose [1994] considers RD bifurcations of continuous source alphabets, usually assuming a squared-error distortion measure. However, it is assumed there [Rose, 1994, IV.C] that the distortion varies continuously with β and that the solution's support grows monotonically with β. As a result, bifurcations of continuous sources are classified there as either "split" or "mass growing" bifurcations. However, both of these assumptions need *not* hold for finite RD problems, as seen by the rightmost bifurcation of Figure 6.2. cf., Sections 6.3 and 6.5. In related contexts, Parker et al. [2010] consider bifurcations using the Lyapunov-Schmidt reduction, a general-purpose tool for handling singularities. In contrast, we exploit the structure of RD problems by using the reductions defined in Section I.3.1. This simplifies the work with RD roots, facilitating the results below.

6.1 Suboptimal RD curves

We proceed with the discussion in Section I.2.3 around the definition (2.20) of BA_β. The Blahut-Arimoto algorithm converges to a curve-achieving distribution, [Csiszár, 1974, Theorem 1], yet in a manner which depends on the choice of initial condition r_0, as hinted by Csiszár. When r_0 is of full support, we have the following.

Theorem 17 (Csiszár [1974]). *Let p_X and $d(x, \hat{x})$ define an RD problem on a finite reproduction alphabet $\hat{\mathcal{X}}$, and let $\beta > 0$. Let $r_0 \in \Delta[\hat{\mathcal{X}}]$ be an initial condition of full support, $r_0(\hat{x}) > 0$ for all $\hat{x} \in \hat{\mathcal{X}}$. Then, there exists a curve-achieving distribution r^* such that $BA_\beta^n[r_0] \xrightarrow{n \to \infty} r^*$.*

Let r_0 be an initial condition of full support, and denote by r^* its limit under BA_β, an achieving distribution. If $\operatorname{supp} r^* := \{\hat{x} \in \hat{\mathcal{X}} : r^*(\hat{x}) > 0\}$ is of size 2 at least, then we can choose a non-empty proper subset $\hat{\mathcal{X}}'$ of $\operatorname{supp} r^*$. Next, invoke Theorem 17 on the RD problem restricted to $\hat{\mathcal{X}}'$, starting at some initial condition r_0' of full-support on $\hat{\mathcal{X}}'$, $r_0' \in \Delta[\hat{\mathcal{X}}']$. This yields an achieving distribution $r^{*\prime}$ which obviously differs from r^*, as their supports differ. While r^* achieves the rate-distortion curve of the problem we have started with, $r^{*\prime}$ achieves that of the restricted problem. These two RD curves may differ, as demonstrated by Figure 6.1. cf., Lemma 1 at [Berger, 1971, Section 2.5].

In light of the above, we have the following refinement of Csiszár's Theorem 17,

Theorem 18. *Under the conditions of Theorem 17, let $r_0 \in \Delta[\hat{\mathcal{X}}]$ be an initial condition, not necessarily of full support. Then, there exists a distribution r^* which achieves the rate-distortion curve of reduced RD problem to $\operatorname{supp} r_0$, such that $BA_\beta^n[r_0] \xrightarrow{n \to \infty} r^*$.*

This shows that RD problems have sub-optimal solution branches. Namely, distributions that obtain the RD curve of a restricted problem, but not that of the unrestricted problem. Proceeding with the argument above Theorem 18, one might expect that many sub-optimal branches typically exist at a given β value. Indeed, this is often the case, as demonstrated by Figure 6.1.
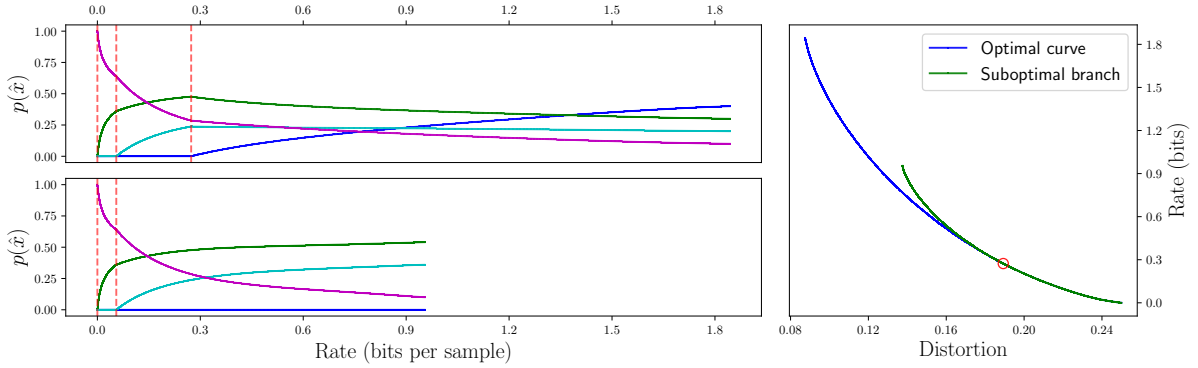

Figure 6.1: **An optimal and a sub-optimal branch merge at a cluster-vanishing bifurcation. Top left**: The optimal solution by rate for the problem in Figure 3.1; BA with uniform initial conditions. **Bottom left**: The solution of the restricted problem, with the blue cluster deleted (equivalently, initialized to zero). **Left**: The solutions of the restricted and unrestricted problems differ above the bifurcation at $R_c \approx 0.273$ (rightmost dashed red vertical), merging into one at the bifurcation point and to its left. **Right**: The rate-distortion curves of both branches intersect at this point, marked by a red circle.

6.2 Cluster-vanishing bifurcations (support shrinking)

The support of an optimal RD solution typically changes with β. A priori, it may not be clear that if the support of an optimal solution changes then there must be a bifurcation. Namely, that the number of fixed points of BA_β must change. We prove that if the support of an optimal solution shrinks, then two roots must intersect and merge into one. The argument for support-switching bifurcations is subtler and deferred to Section 6.5.

When the smallest distortion possible is desired, then the channel $x \mapsto \operatorname{argmin}_{\hat{x}} d(x, \hat{x})$ is clearly optimal; it is often of full support. While for zero rate, the constant encoding to a letter in $\operatorname{argmin}_{\hat{x}} \mathbb{E}[d(X, \hat{x})]$ is optimal. In between, the support of achieving distributions $\boldsymbol{r}_\beta^*$ usually shrinks gradually as a larger distortion D is allowed (β decreased). Suppose that the support of $\boldsymbol{r}_\beta^*$ shrinks at β_c, $\operatorname{supp} \boldsymbol{r}_{\beta-}^* \subsetneq \operatorname{supp} \boldsymbol{r}_{\beta+}^*$. At β_c itself, $\boldsymbol{r}_{\beta_c}^*$ may be considered as a solution of the reduced problem on $\operatorname{supp} \boldsymbol{r}_{\beta-}^*$ by deleting letters $\hat{x}$ outside its support, [Berger, 1971, Lemma 1 in Section 2.5]. By Theorem 11 in Section 5.2, the Jacobian $D_{\boldsymbol{r}} (Id - BA_\beta)|_{\boldsymbol{r}_{\beta_c}^*}$ at the reduced solution is non-singular. Thus, by the Implicit Function Theorem, $Id - BA_\beta$ on the reduced problem has a unique root $\boldsymbol{r}_\beta'$ through $(\boldsymbol{r}_{\beta_c}^*, \beta_c)$. While below β_c it coincides with $\boldsymbol{r}_{\beta-}^*$, above β_c it must differ from $\boldsymbol{r}_{\beta+}^*$ as their supports differ, although both are fixed points of BA_β (2.20). This shows that two roots of $Id - BA_\beta$ must intersect at β_c, as demonstrated to the left of Figure 6.1.

As a side note, we comment that if an achieving distribution $\boldsymbol{r}_\beta^*$ is known to be unique, then $\boldsymbol{r}_\beta'$ is unstable above β_c. For, adding a small non-zero perturbation to the coordinates of $\boldsymbol{r}_\beta'$ will then result in an initial condition which converges to $\boldsymbol{r}_\beta^*$ under BA_β, by Csiszár's Theorem 17.

6.3 Obstructions to the root-tracking assumptions for RD

We discuss the obstructions to Assumptions 1 and 2 (from Section I.2.1) in RD context. Namely, that the achieving distributions $\boldsymbol{r}$ can be written as a smooth function $\boldsymbol{r}_\beta$ of β. We also discuss a simple method to detect such obstructions.

To that end, we dive into the subtle differences between three possible choices: of fixing a multiplier value $\beta > 0$, a point (D, R) on the rate-distortion curve, or a distribution $q(\hat{x}|x)$ which achieves the RD curve. By [Berger, 1971, Theorem 2.5.2], for every point (D, R) on the RD curve there is some β value such that (D, R) can be generated parametrically[13] from β. This justifies talking about "the curve points (D, R) of β", or "the achieving distributions $\boldsymbol{q}$ of β". However, to a value of β there may correspond more than a single point on the rate-distortion curve; and a point on the rate-distortion curve may be achieved by more than a single distribution. We described these two kinds of non-uniquenesses schematically by,

$$\beta \quad \rightsquigarrow \quad \text{curve points } (D, R) \quad \rightsquigarrow \quad \text{achieving distributions } q(\hat{x}|x) \qquad (6.1)$$

For non-uniqueness to the left of (6.1), the multiplier $-\beta$ is the slope of the RD curve (Theorem 2.5.1 in Berger [1971]), and is a continuous function $\beta(D)$ of the distortion D, on the open interval (D_{min}, D_{max}) (Theorem 2.5.5). As $R(D)$ is convex, its slope $-\beta$ is monotonically non-decreasing in D. Thus, $D \in (D_{min}, D_{max})$ can be written as a function of β if and only if $-\beta(D)$ is strictly increasing, in which case $D(\beta)$ is continuous. When $\beta(D)$ is constant in D, the RD curve has an entire linear segment which corresponds to a single β value. In particular, the achieving distributions cannot be written then as a function of β, breaking Assumption 1. e.g., the right bifurcation in Figure 6.2. See [Berger, 1971, Section 2.7] for further examples.

However, even if the RD curve has no linear segments, a distribution $\boldsymbol{q}$ to the right of (6.1) which achieves a particular curve point (D, R) need not be unique. For example, suppose that

---

[13]By Equations (2.5.15) and (2.5.16) there; see also the discussion following Equation (2.5.19) there.

a column in the distortion matrix is duplicate (d is degenerate): there are $\hat{x}_1 \neq \hat{x}_2$ such that $d(\cdot, \hat{x}_1) = d(\cdot, \hat{x}_2)$. The clusters $\hat{x}_1$ and $\hat{x}_2$ are then indistinguishable for any practical purpose. e.g., if r is an achieving distribution with $r_{12} := r(\hat{x}_1) + r(\hat{x}_2) > 0$, then it is not difficult to see that dividing r_{12} arbitrarily between $r(\hat{x}_1)$ and $r(\hat{x}_2)$ also yields an achieving distribution r': for $\lambda \in [0, 1]$, set $r'(\hat{x}_1) := \lambda \cdot r_{12}$, $r'(\hat{x}_2) := (1 - \lambda) \cdot r_{12}$ and $r'(\hat{x}) := r(\hat{x})$ otherwise. Indeed, Berger notes that "if the distortion matrix exhibits certain form of symmetry and degeneracy, there can be many choices of [a minimizer]". While this breaks neither of our assumptions, the solution may then have multiple parameterizations in terms of β.

Non-uniqueness to the right of (6.1) (multiple distributions achieving a single curve point) implies that the simplex contains an entire line of achieving distributions,

Theorem 19 (Theorem 2.4.2 at Berger [1971])**.** *If the conditional probability distributions q' and q'' both achieve a point $(D, R(D))$ on the rate-distortion curve, then so do their convex combinations $\lambda q' + (1 - \lambda)q''$, for any $\lambda \in [0, 1]$.*

We generalize this as follows, to account for any non-uniqueness at (6.1).

Theorem 20. *For any $\beta > 0$ value, the set of achieving distributions which corresponds to β is convex.*

Proof of Theorem 20. For a conditional probability distribution q and $\beta > 0$, denote by $\mathcal{L}_\beta[q]$ the value of the RD Lagrangian there, $I(q; p_X) + \beta \, \mathbb{E}_{q(\hat{x}|x)p_X(x)}[d(x, \hat{x})]$.

Suppose that $q' \neq q''$ both achieve the RD curve (1.1) at the same β value. Therefore, $\mathcal{L}_\beta[q']$ and $\mathcal{L}_\beta[q'']$ must both equal the minimal Lagrangian value $\mathcal{L}_\beta^*$. Let $q_\lambda := \lambda q' + (1 - \lambda)q''$ for $\lambda \in [0, 1]$. We need to prove that q_λ also achieves a point on the RD curve. Since mutual information is convex in q and the expectation term in $\mathcal{L}$ linear, we have that

$$\mathcal{L}_\beta[q_\lambda] \leq \lambda \mathcal{L}_\beta[q'] + (1 - \lambda)\mathcal{L}_\beta[q''] = \mathcal{L}_\beta^* . \tag{6.2}$$

But $\mathcal{L}_\beta^*$ is the minimal value of the Lagrangian, and so equality follows.

Finally, the RD curve is the envelope of lines of slope $-\beta$ and intercept $\mathcal{L}_\beta^*$ along the R-axis, Berger [1971]. Thus, q_λ indeed achieves the curve (1.1). $\quad\square$

Theorem 20 yields two important corollaries. Recall the definition (2.20) of $BA_\beta[r]$ (Section I.2.3), where it is considered as an operator in the marginals r. By abuse of notation, write $BA_\beta[q]$ for the evaluation of BA_β at q, now considered as an operator in the encoders q (see comments there). If for a particular β value there is more than one achieving distribution $q' \neq q''$, then by Theorem 20, the entire line section connecting q' to q'' is comprised of fixed points of BA_β, or equivalently roots of $Id - BA_\beta$. Therefore, its Jacobian with respect to these coordinates must vanish along the vector pointing from q' to q''.

Corollary 21 (Non-uniqueness of RD solutions is detectable by q-Jacobian)**.** *If at $\beta > 0$ there is more than one achieving distribution q, then $\ker D_q(Id - BA_\beta)[q] \neq \{0\}$.*

The Jacobian in these coordinates is given explicitly by Proposition 23 below. An important practical aspect of this Corollary is that the kernel may be calculated at *any* of the problem's achieving distributions q which correspond to β; it does not matter when testing for uniqueness. If r^* is a fixed point of BA when initialized at an arbitrary marginal r_0 (*not* necessarily of full support) then, by Theorem 18, it achieves the curve of the reduced problem to r_0. Thus, $\ker D_q(Id - BA_\beta)[q^*]$ gives a simple tool to test whether there might be additional distributions achieving a reduced problem at the corresponding encoder q^*.

We note that Corollary 21 is not merely a logical negation of the Implicit Function Theorem, as the latter is local in nature. Compare for example to the lines intersecting parabola Example

in Section I.2.1.1. So long that the base β_0 of the expansion there is above the critical point, the problem has two distinct solutions, yet the Jacobian there is non-singular at each. Unlike Corollary 21 for RD, knowledge of one solution in that example does not allow us to detect that the other exits. Instead, this discussion boils down to the following. cf., [Kuznetsov, 2004, Section 2.3].

Corollary 22. *There are no global bifurcations in rate-distortion problems.*

We note, however, that bifurcations may not be detectable unless encoder coordinates q are used, as explained below. The Jacobian's explicit form in these coordinates is given below (proof in Section III.C.5). cf., the Jacobian $D_r \left(Id - BA_\beta \right)[r]$ in marginal coordinates (Corollary 10 in Section 5.2).

Proposition 23. *Let q be a conditional distribution, s the marginal defined by it via the marginal Equation (2.19). Then, the Jacobian of BA_β in encoder coordinates is given by the matrix*

$$(D_q BA_\beta[q])_{(\hat{x},x),(\hat{x}',x')} = \frac{e^{-\beta d(x,\hat{x}')}}{\sum_{\hat{x}''} s(\hat{x}'') e^{-\beta d(x,\hat{x}'')}} \left[\delta_{\hat{x},\hat{x}'} - BA_\beta[q](\hat{x}|x) \right] p_X(x') \tag{6.3}$$

whose rows and columns are indexed by $(\hat{x},x)$ and $(\hat{x}',x') \in \hat{\mathcal{X}} \times \mathcal{X}$, respectively. When $\forall \hat{x} \; s(\hat{x}) \neq 0$, this simplifies to

$$\frac{BA_\beta[q](\hat{x}'|x)}{s(\hat{x}')} \left[\delta_{\hat{x},\hat{x}'} - BA_\beta[q](\hat{x}|x) \right] p_X(x') . \tag{6.4}$$

There is a subtle difference between the Jacobians of $Id - BA_\beta$ with respect to the cluster marginal r and encoder q coordinates. For, the r-Jacobian can only detect cluster-vanishing bifurcations, as explained below, while the q-Jacobian can be used to detect *any* RD bifurcation, by Corollary 21. As a result, it is easy to mis-detect bifurcations other than cluster-vanishing when using marginal coordinates r. e.g., the support-switching bifurcation to the right of Figure 6.2 appears as a discontinuity in marginal coordinates r (panel A), with no Jacobian eigenvalue vanishing to indicate its appearance (panel C). Nevertheless, its encoder coordinates q-Jacobian has an eigenvalue vanishing precisely there (panel D); cf., Section 6.5 on support-switching bifurcations.

By the Implicit Function Theorem, if $D_q(Id - BA_\beta)|_q$ has no kernel at its root q then there is a unique function q_β through it. As discussed in Section I.2.1, this implies that Assumptions 1 and 2 hold. In particular, if q is an achieving distribution, then by Corollary 21 there is no other curve-achieving distribution at the same β value. For, that would be detectable by a non-trivial kernel. cf., the discussion around Equation (6.1).

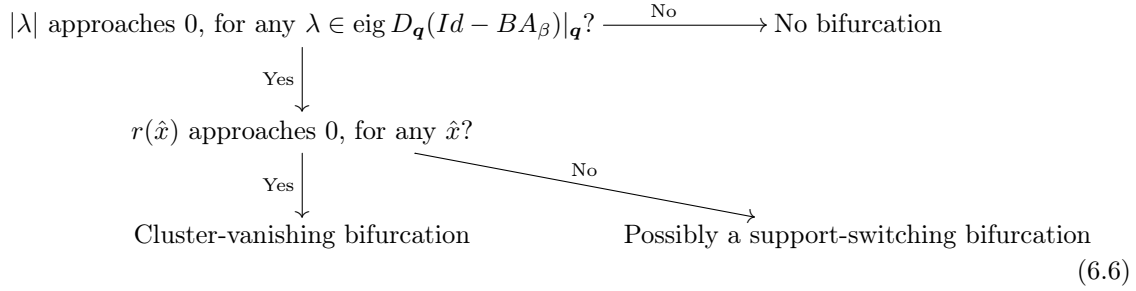**Corollary 24.** *Let $\beta > 0$, and let q be a fixed point of BA_β such that*

$$\ker D_q(Id - BA_\beta)|_q = \{\mathbf{0}\} . \tag{6.5}$$

Then, Assumptions 1 and 2 hold there for RD.

Further, if q is also an achieving distribution, then (6.5) implies that there is no other curve-achieving distribution at that β value.

In contrast, the Jacobian $D_r(Id - BA_\beta)|_r$ in marginal coordinates can only detect bifurcations where the support shrinks to a proper subset, at least in non-degenerate problems. For, by [Agmon et al., 2021, Lemma 2 ff.], its kernel is determined only by $\operatorname{supp} r$ (corresponding precisely to clusters outside the support), regardless of how many other achieving distributions there may be. By reducing the problem to $\operatorname{supp} r$ (as in Section 6.2), its eigenvalues can be seen to be continuous in β, vanishing if and only if a cluster vanishes. Therefore, the r-Jacobian cannot detect a bifurcation that switches between two distinct supports, even if they are of the same size. e.g., the right bifurcation of Figure 6.2.

The above gives a way to distinguish numerically between bifurcations of different types when tracking RD roots, summarized at (6.6) below. Since eigenvalues are continuous in the choice of matrix, and we assume that $D_{\boldsymbol{q}}(Id - BA_\beta)|_{\boldsymbol{q}}$ is usually of full rank (Assumption 3), then one can put a small threshold on eigenvalues, below which the Jacobian is considered singular. This is a necessary condition for bifurcation, of any kind. For simplicity, suppose that the total algebraic multiplicity of the vanishing eigenvalues is 1. If any cluster vanishes simultaneously, then the fixed point is approaching a cluster-vanishing bifurcation. Otherwise, the fixed point may be approaching a bifurcation of some other kind, or the distortion matrix may misbehave, as noted after (6.1). While several approaches may come to mind on how bifurcations other than cluster-vanishing can be handled, that is beyond the scope of this work.

$|\lambda|$ approaches 0, for any $\lambda \in \mathrm{eig}\, D_{\boldsymbol{q}}(Id - BA_\beta)|_{\boldsymbol{q}}$? $\xrightarrow{\quad \text{No} \quad}$ No bifurcation

$\Big\downarrow$ Yes

$r(\hat{x})$ approaches 0, for any $\hat{x}$?

$\Big\downarrow$ Yes $\qquad\qquad \xrightarrow{\quad\quad \text{No} \quad\quad}$

Cluster-vanishing bifurcation $\qquad\qquad$ Possibly a support-switching bifurcation

$$\tag{6.6}$$

The fact that there are RD bifurcations that are captured by the kernel of the $\boldsymbol{q}$-Jacobian but *not* by that of the $\boldsymbol{r}$-Jacobian can be understood intuitively by the relation

$$(D_{\boldsymbol{r}}BA_\beta[\boldsymbol{r}])_{\hat{x},\hat{x}'} = \sum_{x=x'} (D_{\boldsymbol{q}}BA_\beta[\boldsymbol{q}])_{(\hat{x},x),(\hat{x}',x')} \tag{6.7}$$

at a fixed point of full-support, which can be verified directly. That is, $D_{\boldsymbol{r}}BA_\beta[\boldsymbol{r}]$ is the blockwise trace of $D_{\boldsymbol{q}}BA_\beta[\boldsymbol{q}]$, and so it contains only a "summary" of the information in the latter. In particular, it may be that $D_{\boldsymbol{q}}BA_\beta[\boldsymbol{q}]$ has an eigenvector of eigenvalue 1 while $D_{\boldsymbol{r}}BA_\beta[\boldsymbol{r}]$ does not.

As a side note, the argument of Agmon et al. for critical slowing down of Blahut-Arimoto was based on the Jacobian $D_{\boldsymbol{r}}(Id - BA_\beta)|_{\boldsymbol{r}}$ with respect to cluster-marginal coordinates. However, their Theorem 5 makes no use of the choice of coordinates system, and so implies critical slowing down also when an eigenvalue of the $\boldsymbol{q}$-Jacobian vanishes gradually, even if no eigenvalue of the $\boldsymbol{r}$-Jacobian is vanishing. This is demonstrated by Figure 6.2 (panels B and D, right bifurcation).
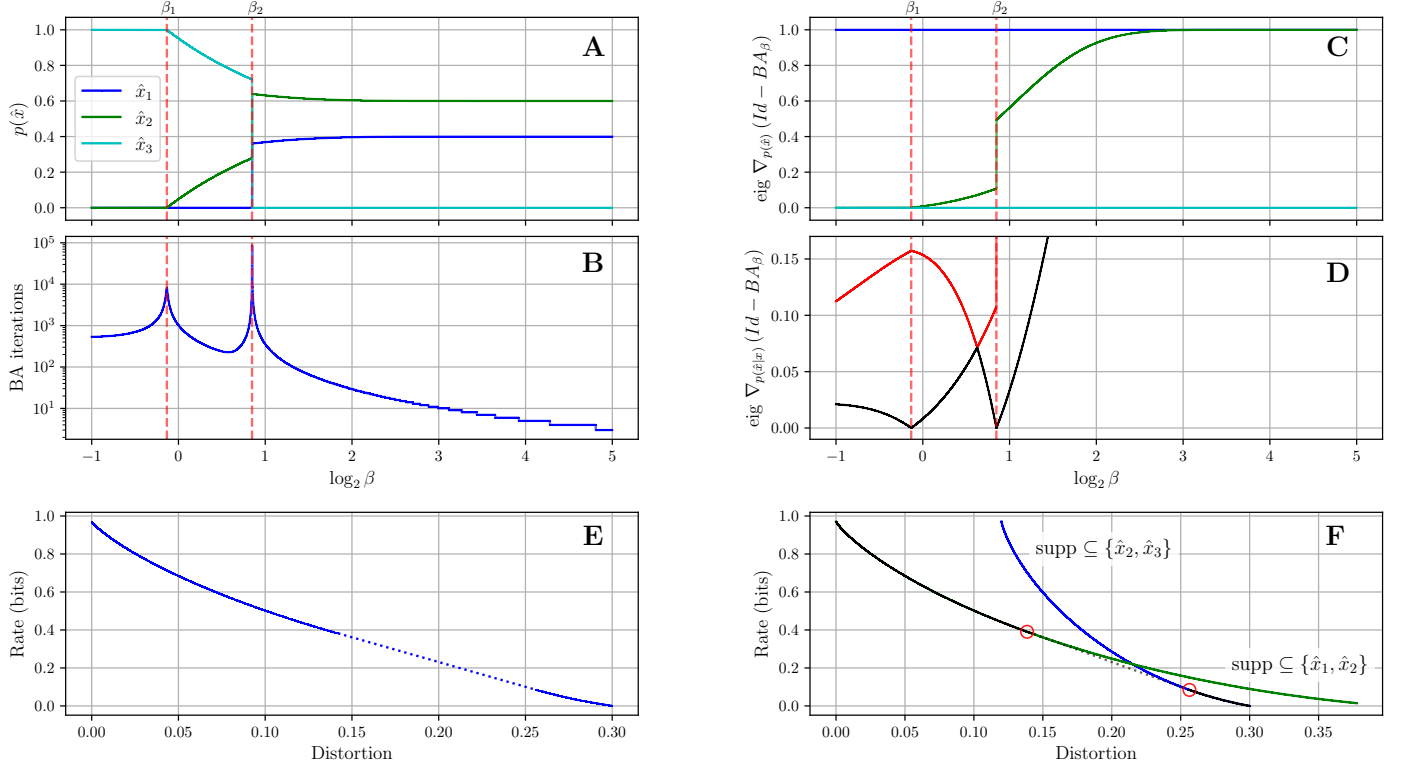
Figure 6.2: **A support-switching and a cluster-vanishing bifurcation.** Reproducing [Berger, 1971, Figure 2.7.6], defined by $d(x, \hat{x}) = \begin{pmatrix} 1 & 0 & 0.3 \\ 0 & 1 & 0.3 \end{pmatrix}$ and $p_X = (0.4, 0.6)$. This problem exhibits two bifurcations, at β_1 and β_2 (dashed red verticals). **A.** Cluster marginal $p_\beta(\hat{x})$ as a function of β. While at β_1 a cluster vanishes, at β_2 the support switches between two distinct subsets of size 2. At β_2, an entire line section of distributions is optimal. This results in an apparent discontinuity at panels A through D, as they are plotted by β value. **B.** Blahut-Arimoto iterations until convergence (stopping condition is 10^{-7}), initialized with uniform initial conditions at each β value. Critical slowing down is clearly noticed at both sides of the two bifurcations, even though panel C has a vanishing eigenvalue only to the right of β_1. This is explained by the vanishing eigenvalues of the $\boldsymbol{q}$-Jacobian with respect to encoder coordinates, at panel D. See Section 6.3. **C.** Eigenvalues of the $\boldsymbol{r}$-Jacobian with respect to cluster-marginal coordinates $r(\hat{x})$. An eigenvalue approaches zero at the cluster-vanishing bifurcation to the left. In contrast, the solution's support size does not change in the vicinity of β_2. Therefore, the Jacobian's rank there remains unchanged [Agmon et al., 2021, Theorem 1]. **D.** Eigenvalues of the $\boldsymbol{q}$-Jacobian with respect to the encoder's coordinates $q(\hat{x}|x)$. Multiple achieving distributions at a single value β_2 are detectable by a non-trivial kernel direction, in accordance with Corollary 21. **E.** The rate-distortion curve has a linear segment (dotted), corresponding to $\beta = \beta_2$. **F.** The linear curve segment can be explained in terms of a support-switching bifurcation between two suboptimal RD curves, of the problems restricted to $\{\hat{x}_1, \hat{x}_2\}$ and $\{\hat{x}_2, \hat{x}_3\}$. The optimal RD curve in black (dashed) alternately coincides with the two suboptimal curves (blue and green), as explained in Section 6.5. Both our Algorithm 3 and BA with reverse annealing miss the right bifurcation, and so follow the sub-optimal root in green; see Section 6.4.

6.4 Why does root-tracking for RD follow the optimal solution path?

As seen in previous subsections, RD problems typically have a plethora of sub-optimal solutions, which do *not* achieve the problem's rate-distortion curve. Under Assumption 3 (of Section I.3.1), the convexity of achieving distributions (Theorem 20) implies that an optimal root tracked by our algorithms does indeed remain optimal (namely, achieving). We elaborate on this below.

First, consider Algorithm 2 for tracking a root between cluster-vanishing bifurcations. An achieving distribution tracked by it remains achieving so long that the $\boldsymbol{q}$-Jacobian is non-singular. Indeed, by Corollary 24 in Section 6.3, Assumptions 1 and 2 which are necessary for root-tracking I.2.2 hold. Further, as the root being tracked is an achieving distribution, then that Corollary also implies that there is no other achieving distribution. We note that two distinct roots cannot exchange the property of being curve achieving without both being achieving simultaneously, because the rate and distortion functionals are continuous in the encoder $\boldsymbol{q}$, [14]. The above arguments hold so long that $D_{\boldsymbol{q}}(Id - BA_{\beta})[\boldsymbol{q}_{\beta}]$ remains non-singular, which by Assumption 3 is true until the next cluster-vanishing bifurcation is reached.

Second, when an achieving distribution $\boldsymbol{q}_{\beta}$ approaches a cluster-vanishing bifurcation, then by continuity of the RD curve, [Berger, 1971, Theorems 2.4.1, 2.5.4], it achieves the curve also at the point of bifurcation. As shown in Section 6.2, a sub-optimal root must exist to the right of the bifurcation, merging with the optimal one at the point of bifurcation. Thus, while the heuristic used by Algorithm 3 to handle the bifurcation may temporarily follow a sub-optimal branch, it achieves the curve once the point of bifurcation is reached, re-gaining optimality.

Third, we need to guarantee that Algorithm 2 is indeed initialized at an achieving distribution. When starting at $\beta_0 \gg 0$ with an initial condition of full support, iterate with Blahut-Arimoto until convergence. By Csiszár's Theorem 17, the distribution $\boldsymbol{q}_{\beta_0}$ obtained this way is an achieving distribution. As shown above, the heuristic used by Algorithm 3 always ends at an achieving distribution. Thus, Algorithm 3 will initialize Algorithm 2 at achieving distribution the next time it is invoked.

The assumption that the $\boldsymbol{q}$-Jacobian is non-singular outside cluster-vanishing bifurcations is necessary for Algorithm 2 to keep track of the optimal root. For, this algorithm tracks the root of the reduced problem, essentially using only the $\boldsymbol{r}$-Jacobian as its stopping condition. Therefore, it would only detect cluster-vanishing bifurcations, as explained in Section 6.3. e.g., at the support-switching bifurcation to the right of Figure 6.2, Algorithm 2 would continue to track the root past the bifurcation, even after it had lost optimality. So will BA with reverse annealing, as the root continues to exist beyond the bifurcation. However, while missing bifurcations other than cluster vanishing, Algorithm 2 would suffer no computational penalty or accuracy loss near them as Blahut-Arimoto does, since no eigenvalue of the $\boldsymbol{r}$-Jacobian vanishes there (cf., panels B and C of Figure 6.2).

Nevertheless, the discussion around (6.6) provides a simple method to detect bifurcations other than cluster-vanishing ones, up to pathologies of the distortion matrix. Once detected, we expect that they could be handled using tools similar to those developed here. As an alternative approach, RD derivatives in Algorithm 1 can be calculated with respect to $\boldsymbol{q}$ rather than $\boldsymbol{r}$ coordinates, changing the stopping condition of Algorithm 2 accordingly. We have not chosen this approach due the computational costs of having a higher-dimensional variable.

---

[14]Denote $I(\boldsymbol{q}) := I(\boldsymbol{q}; p_X)$, $D(\boldsymbol{q}) := \mathbb{E}_{\boldsymbol{q} \, p_X}[d(x, \hat{x})]$, $F_{\beta}(\boldsymbol{q}) := I(\boldsymbol{q}) + \beta D(\boldsymbol{q})$, and write $F_{\beta} := \min_{\boldsymbol{q}} F_{\beta}(\boldsymbol{q})$ for its minimal value. Suppose that $\boldsymbol{q}'_{\beta}$ and $\boldsymbol{q}''_{\beta}$ are (continuous) paths such that $F_{\beta_1}(\boldsymbol{q}'_{\beta_1}) < F_{\beta_1}(\boldsymbol{q}''_{\beta_1})$ but $F_{\beta_2}(\boldsymbol{q}'_{\beta_2}) > F_{\beta_2}(\boldsymbol{q}''_{\beta_2})$, for some $\beta_1 < \beta_2$. Then by continuity, there must exist $\beta_3 \in (\beta_1, \beta_2)$ where $F_{\beta}(\cdot)$ obtains the same value on both paths, $F_{\beta_3}(\boldsymbol{q}'_{\beta_3}) = F_{\beta_3}(\boldsymbol{q}''_{\beta_3})$. The argument follows by setting $F_{\beta_1} = F_{\beta_1}(\boldsymbol{q}'_{\beta_1})$ and $F_{\beta_2} = F_{\beta_2}(\boldsymbol{q}''_{\beta_2})$.

6.5 Linear curve segments as support-switching bifurcations

As a side note to this Section's main line of discussion, we offer an explanation of linear curve segments in terms of support-switching bifurcations between sub-optimal RD curves. This explains the right bifurcation in Figure 6.2.

Suppose that at D_1 the curve is achieved by a solution of support $A \subseteq \hat{\mathcal{X}}$, and at $D_2 > D_1$ by a solution of support $B \subseteq \hat{\mathcal{X}}$, where neither of these subsets contains the other, $A \nsubseteq B$ and $B \nsubseteq A$. The case of a shrinking support at bifurcation was already handled in Section 6.2. Without loss of generality, suppose that $\hat{\mathcal{X}} = A \cup B$. For simplicity, suppose that outside bifurcations, the curve is achieved by a unique distribution. Denote by $R^A(D)$ and $R^B(D)$ the RD curves of the restricted problems to A and B, respectively, and suppose that their respective ranges (D_{min}^A, D_{max}^A) and (D_{min}^B, D_{max}^B) of distortion values has a large enough intersection for the below to be meaningful.

Since both $R^A(D)$ and $R^B(D)$ are continuous, and only R^A is optimal at D_1 (only R^B at D_2), then the two curves must intersect somewhere in (D_1, D_2). Recall that a differentiable function is convex if and only if all its tangents lie below its graph. Thus, the tangent to R^A at a point D is below the curve of R^A. Pick the smallest D_1' such that the tangent to R^A at D_1' also intersects the curve R^B, say at D_2'. There must be such a point D_1', as the two curves R^A and R^B intersect and are continuous. $D_1' > D_1$ because R^A is below R^B at D_1, by assumption. By convexity, $D_1 < D_1' < D_2' < D_2$. If R^A still coincides with the problem's RD curve at D_1' and R^B with that at D_2', then by Theorem 20, the entire line section connecting $(D_1', R^A(D_1'))$ to $(D_2', R^A(D_2'))$ is achievable, at the β value corresponding to its slope. Other than its vertices at D_1' and D_2', which are obtained by distributions supported on A or B alone, any other point along this section is obtained by distributions supported on the entire $\hat{\mathcal{X}}$. Points in the section's interior lie on the RD curve of the original problem, but on neither of the suboptimal curves R^A or R^B. See panel F in Figure 6.2 for example.

7 Error analysis for root-tracking for RD

We analyze the error of Algorithm 2, our specialization of Taylor's method to RD. While the highlight of this section is its convergence guarantees (Theorem 5 in Section I.3.1; see Section III.D.2 for proof), the results comprising it are of interest on their own right. In Section 7.1 we briefly recap standard error analysis, highlighting relevant subtleties. In Section 7.2 we show that tracking an operator's root in the presence of bifurcations generally belongs to a family of "stiff" problems, which are harder to solve with standard numerical methods such as Taylor's. Stopping towards a bifurcation effectively restricts the problem's difficulty, allowing RD derivative tensors to be bounded uniformly. To mitigate this difficulty, we suggest in Section 7.3 a tool for local error estimation, which could be used to improve the cost-to-error tradeoff of Algorithm 2.

7.1 Preliminaries: error analysis of Taylor methods

We succinctly recap the error analysis of the Euler and Taylor methods for solving ordinary differential equations. While these are necessary for the sequel, the reader well versed in this material is advised to skip to the next subsection.

We follow standard definitions of numerical approximations, as in Atkinson et al. [2011] or Butcher [2016]. A first-order initial value problem is defined by

$$\frac{d\boldsymbol{x}}{d\beta} = f(\boldsymbol{x}, \beta), \quad \boldsymbol{x}(\beta_0) = \boldsymbol{x}_0 . \tag{7.1}$$

By the implicit ODE (2.4), tracking an operator root $(\boldsymbol{x}_0, \beta_0)$ is of this form. Where, RD roots satisfy the ODE of Theorem 14 in Section 5.3.

Write $\boldsymbol{x}_n$ for a numerical approximation of the true solution $\boldsymbol{x}(\beta_n)$ at $\beta := \beta_n$, and $\boldsymbol{e}_n := \boldsymbol{x}(\beta_n) - \boldsymbol{x}_n$ for the approximation's error, known as the *global truncation error* in the context of numerical approximation. A numerical approximation can be obtained by setting

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n + \Delta\beta \cdot f\left(\boldsymbol{x}_n, \beta_n\right) \tag{7.2}$$

where $\Delta\beta$ is the *step size*, often fixed, and $\beta_{n+1} := \beta_n + \Delta\beta$. This approximation method (7.2) is known as the *Euler method*. When its right-hand side is replaced by a Taylor polynomial of degree $l > 0$, this is known as the *Taylor method*,

$$\boldsymbol{x}_{n+1} := \boldsymbol{x}_n + \Delta\beta \cdot T_l\left(\boldsymbol{x}_n, \beta_n, \Delta\beta\right) . \tag{7.3}$$

Euler's method is a first-order Taylor method. In root-tracking context, we write

$$T_l\left(\boldsymbol{x}_n, \beta_n, \Delta\beta\right) := \frac{1}{1!} \cdot \left.\frac{d\boldsymbol{x}}{d\beta}\right|_{(\boldsymbol{x}_n, \beta_n)} + \frac{\Delta\beta}{2!} \cdot \left.\frac{d^2\boldsymbol{x}}{d\beta^2}\right|_{(\boldsymbol{x}_n, \beta_n)} + \cdots + \frac{\Delta\beta^{l-1}}{l!} \cdot \left.\frac{d^l\boldsymbol{x}}{d\beta^l}\right|_{(\boldsymbol{x}_n, \beta_n)} \tag{7.4}$$

Where, $\frac{d^k\boldsymbol{x}}{d\beta^k}$ are the implicit derivatives computed by Algorithm 1 (Theorem 3). This gives a numerical algorithm for tracking operator roots (1.3), under Assumptions 1 and 2 (in Section I.2.1). Where, in RD context the step $\Delta\beta$ is negative (see Section I.3.1).

Error analysis of Euler's method (7.2) is a standard result in numerical solution of ODEs. e.g., [Atkinson et al., 2011, Theorem 2.4] or [Butcher, 2016, Theorem 212A]. Its generalization to an *l*-th order Taylor method is straightforward, though usually not given explicitly in textbooks. Following the notes of Gottlieb [2006], its crux is subtracting the numerical approximation $\boldsymbol{x}_{n+1}$ (7.3) from the true solution

$$\boldsymbol{x}(\beta_{n+1}) = \boldsymbol{x}(\beta_n) + \Delta\beta \cdot T_l\left(\boldsymbol{x}(\beta_n), \beta_n, \Delta\beta\right) + \Delta\beta \cdot \underbrace{\frac{\Delta\beta^l}{(l+1)!} \frac{d^{l+1}\boldsymbol{x}}{d\beta^{l+1}}}_{LTE} , \tag{7.5}$$

where the last term is the *local truncation error*, which is simply the Taylor remainder at $(\boldsymbol{x}(\beta'), \beta')$, for some intermediate $\beta' \in [\beta_n, \beta_{n+1}]$. This yields

$$\|\boldsymbol{e}_{n+1}\|_\infty \leq \|\boldsymbol{e}_n\|_\infty + |\Delta\beta| \cdot \|T_l\left(\boldsymbol{x}(\beta_n), \beta_n, \Delta\beta\right) - T_l\left(\boldsymbol{x}_n, \beta_n, \Delta\beta\right)\|_\infty + |\Delta\beta| \cdot \|LTE\|_\infty , \tag{7.6}$$

where $\|\cdot\|_\infty$ denotes the supremum norm. If T_l satisfies the *Lipschitz condition with a constant L_l* with respect to $\boldsymbol{x}$, $\|T_l(\boldsymbol{x}, \beta) - T_l(\boldsymbol{x}', \beta)\|_\infty \leq L_l \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$ for every $\boldsymbol{x}, \boldsymbol{x}'$ and $\beta \in [\beta_0, \beta_f]$, then

$$\|\boldsymbol{e}_{n+1}\|_\infty \leq (1 + |\Delta\beta|L_l) \cdot \|\boldsymbol{e}_n\|_\infty + |\Delta\beta| \cdot \|LTE\|_\infty , \tag{7.7}$$

and so the bound grows exponentially. This allows one to show [Atkinson et al., 2011, Equation (5.11)] that the global truncation error of an *l*-th order Taylor method is of order $O(|\Delta\beta|^l)$:

Theorem 25 (Error analysis of Taylor method). *For an initial-value problem (7.1) on $[\beta_0, \beta_f]$, the global truncation error obtained by a Taylor method of order l with a step size of $|\Delta\beta|$ at most satisfies*

$$\max_{\beta_0 \leq \beta_n \leq \beta_f} \|\boldsymbol{x}(\beta_n) - \boldsymbol{x}_n\|_\infty \leq e^{(\beta_f - \beta_0)L_l} \|\boldsymbol{e}_0\|_\infty + \frac{e^{(\beta_f - \beta_0)L_l} - 1}{L_l} \cdot \frac{1}{(l+1)!} |\Delta\beta|^l \max_{\beta_0 \leq \beta \leq \beta_f} \left\|\frac{d^{l+1}\boldsymbol{x}(\beta)}{d\beta^{l+1}}\right\|_\infty \tag{7.8}$$

where L_l is the Lipschitz constant of T_l (7.4), and $\boldsymbol{e}_0 := \boldsymbol{x}(\beta_0) - \boldsymbol{x}_0$ is the initial error.

We note that the Lipschitz constant L_l may be taken [Atkinson et al., 2011, Equation (3.9)] as the supremum of the problem's linearization, $\sup \|D_{\boldsymbol{x}} T_l\|_\infty$, over the relevant domain in $\boldsymbol{x}$ and β. The matrix norm $\|D_{\boldsymbol{x}} T_l\|_\infty$ at a point can be used to estimate the *local Lipschitz constant* of T_l; namely, its Lipschitz constant over an arbitrarily small neighborhood.

7.2 The computational difficulty in root tracking for RD

There is a computational difficulty in tracking operator roots with Taylor's method, a difficulty that stems from the presence of bifurcations. For Taylor's method error analysis (Theorem 25), this is manifested in general via local Lipschitz constants. For, its error bounds explode when approaching a bifurcation. The below Lemma 26 bounds RD derivative tensors, and Proposition 27 (in Section 7.3) can be used to bound local Lipschitz constants. Using both, one can show that setting a cluster-vanishing threshold (as in Algorithm 2) restricts the computational difficulty.

In its implicit ODE form, tracking an operator's root is defined by the initial value problem consisting of

$$\frac{d\boldsymbol{x}}{d\beta} = -\left(D_{\boldsymbol{x}}F\right)^{-1} D_\beta F \ , \tag{2.4}$$

at a given root $(\boldsymbol{x}_0, \beta_0)$ of $F = \boldsymbol{0}$ (1.3), so long that $D_{\boldsymbol{x}}F$ is non-singular. Suppose that an eigenvalue of $D_{\boldsymbol{x}}F$ vanishes gradually as $\beta \to \beta_c$. e.g., when approaching a bifurcation. The Jacobian $D_{\boldsymbol{x}}F$ then usually becomes ill-conditioned as a result[15]. The linearization of this differential equation would then in general be ill-conditioned[16], implying that it is *stiff* [Atkinson et al., 2011, Chapter 8]. While there is no widely accepted definition of stiff equations, their *"most important common feature [...] is that when such equations are being solved with standard numerical methods, the step-size $|\Delta\beta|$ must be extremely small in order to maintain stability — far smaller than would appear to be necessary from a consideration of the truncation error"*, [Atkinson et al., 2011, Chapter 8]. See also [Butcher, 2000, Section 6].

Indeed, while a finite Lipschitz constant is required for Taylor method's error analysis (Theorem 25 above), it need not be bounded near a bifurcation[17]. cf., its error bound (7.7). The existence of a bifurcation is not just a technical hurdle in proving that the Theorem's conditions hold, but an essential one, impeding algorithms' performance there. For Blahut-Arimoto's algorithm, this is manifested by critical slowing down near bifurcations, Agmon et al. [2021], while for tracking an operator's root it is manifested in the stiffness of the implicit ODE (2.4). Both BA and our Algorithm 1 for RD suffer from reduced accuracy when approaching a bifurcation, as depicted by Figure 2.3.

While there is much literature on stiff differential equations, stopping at a cluster mass threshold $\delta > 0$ as in Algorithm 2 is a straightforward solution for guaranteeing convergence of Taylor's method. The proof of Theorem 5 bounds the local Lipschitz constants so long that the bifurcation is at least δ-far. To show this, Lemma 26 below guarantees that RD derivative tensors are then bounded uniformly. While Proposition 27 in Section 7.3 allows to bound not only the implicit derivatives, but also the local Lipschitz constants. Writing $\boldsymbol{r}$ for a distribution in $\Delta[\hat{\mathcal{X}}]$, we have the following.

Lemma 26 (RD derivative tensors are bounded uniformly on compact subsets in simplex interior)**.** *For any $\delta > 0$ small enough, the derivative tensors of $Id - BA_\beta$ (1.2) ((2.30) and (2.31) in Theorem 4) are bounded uniformly on the closed δ-interior of the simplex, under the supremum norm.*

Explicitly, let an RD problem be defined by p_X and $d(x, \hat{x})$ (d for short), and let $\boldsymbol{r}$ be a distribution in the closed δ-interior of $\Delta[\hat{\mathcal{X}}]$. Then, for any orders $b, m \geq 0$ of differentiation (other

---

[15]Unless all its eigenvalues vanish at the same rate when $\beta \to \beta_c$.

[16]Write $D_{\boldsymbol{x}}F^{-1}$ for the inverse of the Jacobian matrix, and differentiate $D_{\boldsymbol{x}}F \ D_{\boldsymbol{x}}F^{-1} = I$ with respect to $\boldsymbol{x}$, to obtain $D_{\boldsymbol{x}}F \ D_{\boldsymbol{x}}(D_{\boldsymbol{x}}F^{-1}) = -D^2_{\boldsymbol{x},\boldsymbol{x}}F \ D_{\boldsymbol{x}}F^{-1}$, or equivalently $D_{\boldsymbol{x}}(D_{\boldsymbol{x}}F^{-1}) = -D_{\boldsymbol{x}}F^{-1} \ D^2_{\boldsymbol{x},\boldsymbol{x}}F \ D_{\boldsymbol{x}}F^{-1}$. This shows that the linearization $-D_{\boldsymbol{x}}\left(D_{\boldsymbol{x}}F^{-1}D_\beta F\right)$ of (2.4) can be written as $-D_{\boldsymbol{x}}F^{-1}M$ for $M := D^2_{\beta,\boldsymbol{x}}F - D^2_{\boldsymbol{x},\boldsymbol{x}}F \ D_{\boldsymbol{x}}F^{-1}D_\beta F$ a matrix. So, it would be ill-conditioned if an eigenvalue λ_β of $D_{\boldsymbol{x}}F$ vanishes as $\beta \to \beta_c$, unless no column of M has a component in the λ_β-eigenspace of $D_{\boldsymbol{x}}F$.

[17]By the argument above, the linearization of (2.4) need not have a finite matrix norm when an eigenvalue of $D_{\boldsymbol{x}}F$ vanishes gradually.

than $b = m = 0$),

$$\left| \left(D^{b+m}_{\beta^b, \boldsymbol{r}^m} (Id - BA_\beta) [\boldsymbol{r}] (\hat{x}) \right)_{(i_1, i_2, \ldots, i_m)} \right| \le 1 + \frac{1}{\delta^m} \cdot C(b, m; d, |\hat{\mathcal{X}}|) \tag{7.9}$$

where

$$C(b, m; d, M) := 2b \cdot (m+1)! \binom{b+M-1}{b} \left[m! \, p(b) \cdot \left(2^b b! \cdot d_{max}^{b^2} \right)^{1+m} \right]^M , \tag{7.10}$$

with $d_{max} := \max\{1, \max_{x, \hat{x}} d(x, \hat{x})\}$, and $p(m)$ is the partition function (see Section I.2.2).

While the bound at (7.9) can be improved, its purpose is to bound the derivative tensors of $Id - BA_\beta$ (1.2) uniformly, with constants involving only the orders b and m of differentiation, and the problem's definition. See Section III.D.1 for its proof. Note that the bound does *not* depend on the variable β, even though BA_β and its fixed points do depend on β. For a derivative with respect to β alone, $m = 0$, the bound is uniform on the entire simplex.

7.3 Local Lipschitz constants of high-order implicit derivatives

The previous Subsection 7.2 shows that it need not be possible to bound Taylor method's local Lipschitz constants uniformly. In the same token, using a fixed order and step size in Algorithm 2 is computationally inefficient. cf., Figure 3.2, and Section I.3.4 for improvements. For both purposes, it is useful to have an estimate of local Lipschitz constants, which we provide below.

The computational inefficiency due to fixed order and step size (Section I.3.4) can be traced back to the variations in local Lipschitz constants, which explode at cluster-vanishing bifurcations. Fitting a single value to all β and $\boldsymbol{x}$ values is too conservative. Instead, one could use $\|D_{\boldsymbol{x}} T_l\|_\infty \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$ to estimate an upper bound to $\|T_l(\boldsymbol{x}, \beta) - T_l(\boldsymbol{x}', \beta)\|_\infty$, if $\boldsymbol{x}$ and $\boldsymbol{x}'$ close enough. This leads to a local estimate of the error's growth rate as in (7.7), up to replacing L_l there with the matrix norm $\|D_{\boldsymbol{x}} T_l|_{(\boldsymbol{x}_n, \beta_n)}\|_\infty$. From the definition (7.4) of T_l, it is a sum of implicit derivative vectors $\frac{d^m \boldsymbol{x}}{d\beta^m}$ for $m = 1, \ldots, l$. And so, to calculate the latter matrix norm it suffices to calculate the Jacobian matrices $D_{\boldsymbol{x}} \frac{d^m \boldsymbol{x}}{d\beta^m}$. A direct calculation in Section III.B.2 yields the formula below. While it involves many more summands than formula (2.17) for implicit derivatives (Theorem 3), all the ingredients needed for $D_{\boldsymbol{x}} \frac{d^l \boldsymbol{x}}{d\beta^l}$ were already computed when calculating $\frac{d^l \boldsymbol{x}}{d\beta^l}$ (if $l \ge 2$).

Proposition 27. *Under the conditions of Theorem 3 (Section I.2.2), suppose further that the Jacobian matrix $D_{\boldsymbol{x}} F$ is invertible. Then, the Jacobian of the l-th order derivative is given by,*

$$
\begin{aligned}
D_{\boldsymbol{x}} \frac{d^l \boldsymbol{x}}{d\beta^l} = & - (D_{\boldsymbol{x}} F)^{-1} \left(D^2_{\boldsymbol{x}, \boldsymbol{x}} F \right) \frac{d^l \boldsymbol{x}}{d\beta^l} \\
& - (D_{\boldsymbol{x}} F)^{-1} \sum_{\substack{non\text{-}trivial \\ partitions}} \sum_{b=0}^{m_1 \cdot \delta(p_1 = 1)} \frac{l!}{b!(m_1 - b)! m_2! \cdots m_s! \cdot (p_1!)^{m_1} \cdots (p_s!)^{m_s}} \\
& \cdot \left\{ D^m_{\beta^b, \boldsymbol{x}^{m-b+1}} F\left[\left(\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}} \right)_{\times (m_1 - b)}, \left(\frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}} \right)_{\times m_2}, \ldots, \left(\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}} \right)_{\times m_s} \right] \right. \\
& + (m_1 - b) \cdot D^m_{\beta^b, \boldsymbol{x}^{m-b}} F\left[D_{\boldsymbol{x}} \frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}}, \left(\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}} \right)_{\times (m_1 - b - 1)}, \left(\frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}} \right)_{\times m_2}, \ldots, \left(\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}} \right)_{\times m_s} \right] \\
& + m_2 \cdot D^m_{\beta^b, \boldsymbol{x}^{m-b}} F\left[\left(\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}} \right)_{\times (m_1 - b)}, D_{\boldsymbol{x}} \frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}}, \left(\frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}} \right)_{\times (m_2 - 1)}, \ldots, \left(\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}} \right)_{\times m_s} \right] \\
& + \ldots \\
& + \left. m_s \cdot D^m_{\beta^b, \boldsymbol{x}^{m-b}} F\left[\left(\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}} \right)_{\times (m_1 - b)}, \left(\frac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}} \right)_{\times m_2}, \ldots, D_{\boldsymbol{x}} \frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}, \left(\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}} \right)_{\times (m_s - 1)} \right] \right\}
\end{aligned} \tag{7.11}
$$

where the summation is over non-trivial integer partitions $(m_1) \cdot p_1 + \cdots + (m_s) \cdot p_s$ of l, and $m := m_1 + \cdots + m_s$.

This proposition is the last building block needed for the proof of Theorem 5, whose proof is brought at Section III.D.2. As with the implicit derivatives $\frac{d^l \boldsymbol{x}}{d\beta^l}$, the Jacobians $D_{\boldsymbol{x}} \frac{d^l \boldsymbol{x}}{d\beta^l}$ essentially contain $(D_{\boldsymbol{x}} F)^{-1}$ to the $(l+1)$-st power, due to the first term at (7.11), and so lose their accuracy when approaching a bifurcation. cf., the notes after Theorem 3 (in Section I.2.2).

8   Complexity of root-tracking and root-tracking for RD

In this section, we present the main complexity results: of root-tracking and root-tracking for RD. We provide bounds for root-tracking both with and without tensor memorization. For RD, $F = Id - BA_\beta$ (1.2), the complexities of the derivatives tensors are broken down to their components, in Table 2 below. Adding these to the complexities of root-tracking yields the complexity bounds of RD root tracking (Theorem 6 in Section I.3.3). See Section III.E for proofs of the below.

Recall, $p(n)$ is the number of partitions of an integer n, with no restriction. One may restrict the number of parts of which a partition is comprised (its total multiplicity). We write $p_{\leq k}(n)$ for the number of partitions when no more than k parts are allowed. e.g., $p_{\leq n}(n) = p(n)$ follows directly from the definition. See Equation (E.4) ff. in Section E.1 for details. Write $C(b, m)$ for the computational complexity of a derivatives tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$. Where, $F(\cdot, \beta)$ is an unspecified operator on $\mathbb{R}^T$, as in Section I.2.1. With this, the complexity of Algorithm 1 for computing implicit derivatives at an operator's root (Section I.2.2) is as follows (proofs in Section E.1).

Proposition 28 (Complexity of l-th order implicit derivative). *Assume that the derivatives $\frac{d^k \boldsymbol{x}}{d\beta^k}$ are known for all $0 < k < l$, and let $C(b, m)$ the complexity of a derivatives tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$. Then, the computational complexity of the l-th order implicit derivative formula (2.17) for $\frac{d^l \boldsymbol{x}}{d\beta^l}$ is*

$$O(T^3) + O\Big(T \cdot \sum_{j=1}^{l} p(j)\Big) + \sum_{m=0}^{l} \sum_{b=0}^{m} p_{\leq m-b}(l-m)\Big[O((m-b+1)T^{m-b+1}) + C(b, m)\Big] \qquad (8.1)$$

The last summand in (8.1) with $C(b, m)$ stands for the cost of computing the derivative tensors, the third for evaluating the multilinear forms they define, the second for summing over the evaluated forms, and the first for finding a linear pre-image under $D_{\boldsymbol{x}} F$.

As seen already by the first few expansion orders (2.8)-(2.10) of $\frac{d^k F}{d\beta^k} = 0$ (in Section I.2.1), derivative tensors are often re-used after their first appearance. Hence, it makes sense to memorize computed tensors so that they are computed only once. This is especially true when the computational costs $C(b, m)$ of derivative tensors $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ are high, as in rate distortion problems. cf., the example in Section I.2.1.1 in contrast.

Proposition 29 (Complexity of l-th order implicit derivative, with tensor memorization). *Assume that the derivatives $\frac{d^k \boldsymbol{x}}{d\beta^k}$ are known for all $0 < k < l$, and let $C(b, m)$ the complexity of calculating a derivative tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$. Assume further that all the derivative tensors $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ with $m < l$ have already been computed. Then, the computational complexity of formula (2.17) for $\frac{d^l \boldsymbol{x}}{d\beta^l}$ is*

$$O(T^3) + O\Big(T \cdot \sum_{j=1}^{l} p(j)\Big) + \sum_{m=0}^{l} \sum_{b=0}^{m} p_{\leq m-b}(l-m) O((m-b+1)T^{m-b+1}) + \sum_{b=0}^{l} C(b, l) \qquad (8.2)$$

The memory complexity of storing all the derivative tensors $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ with $0 \leq b \leq m < l$ is

$$O\Big(l \cdot T^l\Big) \qquad (8.3)$$

Comparing the computational complexity (8.2) to its counterpart (8.1) without tensor memorization (Proposition 28), the number of tensors computed for the l-th order derivative is sliced from $\sum_{j=1}^{l} p(j)$ to just the $l+1$ newly needed tensors. cf., Corollary 37 in Section III.E.1.

Finally, the computational complexity of *all* the implicit derivatives up to order L (including) can be bounded as following:

Proposition 30 (Cumulative complexity of implicit derivatives, with tensor memorization)**.** *Under the conditions of Proposition 29, the computational complexity of all the implicit derivatives* $\frac{d^k \boldsymbol{x}}{d\beta^k}$ *for* $0 < k \leq L$ *is*

$$O\left(e^{9/4 \cdot \ln L + (L+1)\ln T + \pi \sqrt{2L/3}}\right) + \sum_{l=1}^{L} \sum_{b=0}^{l} C(b, l) \,, \tag{8.4}$$

when $L \geq 2$. *When* $L = 1$, *it is*

$$O\left(T^3\right) + C(0, 1) + C(1, 1) \,. \tag{8.5}$$

The memory complexity is as at (8.3)*.*

For RD, our operator is the M-dimensional $Id - BA_\beta$ (1.2) (namely, $T = M$), defined via BA_β (2.20) in marginal coordinates (Section I.2.3). Where, $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$ are the source and reproduction alphabet sizes of the given RD problem. Now that we have complexity results for root tracking, Table 2 summarizes the complexity of computing RD derivative tensors (see Section III.E.2). Combining the complexities of both (in Section III.E.3) yields the complexity bounds for RD root tracking at Theorem 6 (Section I.3.3). We comment that the hyper-exponential terms $2^k k!$ in Table 2 are only exponential in practice. These result from the loose bounds we have provided for the algebraic properties of the P_k polynomials (2.23)-(2.24). In particular, the bounds provided here depend the dimensions of the RD problem at hand, but *not* on its details.

Quantity	Computations	Memory
P_k (2.23)-(2.24)	Irrelevant	$O\left(2^k k!(k+1)\log_2(k+1)\right)$
$\mathbb{E}_{q(\hat{x}'\|x)}\left[d(x, \hat{x}')^k\right]$ (2.28)	$O(MN)$	$O(N)$
$P_k[\boldsymbol{q}; d]$ (2.29)	$O(MNk\, 2^k k!)$	$O(MN)$
$G(k, a)$ (2.32)	$O(MNL \sum_{k=0}^{L} p(k))$	$O(MN(L+1)(L+2))$
$D_{\beta^b}^{b}(Id - BA_\beta)[\boldsymbol{r}]$ (2.30)	$O(MN)$	$O(M)$
$D_{\beta^b, \boldsymbol{r}^m}^{b+m}(Id - BA_\beta)[\boldsymbol{r}]$ (2.31)	$O\left(\binom{m+M-1}{m}\binom{b+M-1}{b} \cdot M^2 N\right)$ $+ O\left(\binom{m+M-1}{m}m!M\right)$	$O(M^{m+1})$

Table 2: **Breakdown of the complexity of RD derivative tensors**. For implicit derivatives up to order L (including), one needs to compute P_k, $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x, \hat{x}')^k\right]$ and $P_k[\boldsymbol{q}; d]$ for $k \leq L$; and the matrices $G(k, a)$. The derivative tensors for $b \leq L$ and for $b + m \leq L$ can then be computed. The hyper-exponential terms $2^k k!$ at P_k and at $P_k[\boldsymbol{q}; d]$ are very loose, and are roughly exponential in practice. For $k \leq 25$, the memory needed to store P_k is roughly $O(1.73^k)$, while the computational complexity of $P_k[\boldsymbol{q}; d]$ is roughly $O(MNk\, 1.56^k)$; see E.2.1 and E.2.2 for details.

Part III
Proofs and technical details

A   Calculations for the line-intersecting-parabola example

We elaborate on the calculations for the example in 2.1.1, in Part I.

Write $\boldsymbol{p} := (x, y)^t$ for the coordinates, $\frac{d\boldsymbol{p}}{d\beta}$ for the vector of derivatives $(\frac{dx}{d\beta}, \frac{dy}{d\beta})^t$. For $i = 1, 2$ and vectors $\boldsymbol{u} = (u_x, u_y), \boldsymbol{v} = (v_x, v_y)$,

$$D_{\boldsymbol{p},\boldsymbol{p}}F_i[\boldsymbol{u}, \boldsymbol{v}] = \frac{\partial^2 F_i}{\partial x \partial y} \cdot (u_x v_y + u_y v_x) + \frac{\partial^2 F_i}{\partial x^2} \cdot u_x v_x + \frac{\partial^2 F_i}{\partial y^2} \cdot u_y v_y \tag{A.1}$$

From the definition (2.11), $\frac{\partial^2 F_1}{\partial x^2} = 2b$ is the only derivative of F at (A.1) which does not vanish, and so

$$D_{\boldsymbol{p},\boldsymbol{p}}F[\tfrac{d\boldsymbol{p}}{d\beta}, \tfrac{d\boldsymbol{p}}{d\beta}] = \left(\frac{dx}{d\beta}\right)^2 \cdot \binom{2b}{0}, \tag{A.2}$$

where $\frac{dx}{d\beta}$ is the first coordinate of $\frac{d\boldsymbol{p}}{d\beta}$. Similarly, $D_{\boldsymbol{p},\boldsymbol{p}}F[\tfrac{d^2\boldsymbol{p}}{d\beta^2}, \tfrac{d\boldsymbol{p}}{d\beta}] = \frac{dx}{d\beta}\frac{d^2x}{d\beta^2} \cdot (2b, 0)^t$. Thus, the first few expansion orders (2.8)-(2.10) around $(x_0, y_0; \beta_0)$ are,

$$0 = \begin{pmatrix} 2bx_0 + c & -1 \\ a & -1 \end{pmatrix} \frac{d\boldsymbol{p}}{d\beta} + \binom{0}{1} \tag{A.3}$$

$$0 = \begin{pmatrix} 2bx_0 + c & -1 \\ a & -1 \end{pmatrix} \frac{d^2\boldsymbol{p}}{d\beta^2} + \left(\frac{dx}{d\beta}\right)^2 \cdot \binom{2b}{0} \tag{A.4}$$

$$0 = \underbrace{\begin{pmatrix} 2bx_0 + c & -1 \\ a & -1 \end{pmatrix}}_{D_{\boldsymbol{p}}F} \frac{d^3\boldsymbol{p}}{d\beta^3} + 3\frac{dx}{d\beta}\frac{d^2x}{d\beta^2} \cdot \binom{2b}{0} \tag{A.5}$$

while the other derivative tensors at (2.8)-(2.10) vanish.

The Jacobian $D_{\boldsymbol{p}}F$ is invertible whenever its determinant does not vanish, which is to say that the slope $2bx_0 + c$ of the parabola at the intersection point differs from the slope a of the line. Its inverse is then $(D_{\boldsymbol{p}}F)^{-1} = \frac{1}{\Delta}\begin{pmatrix} 1 & -1 \\ a & -2bx_0 - c \end{pmatrix}$, where $\Delta := 2bx_0 + c - a$. A straightforward calculation yields,

$$\frac{d\boldsymbol{p}}{d\beta} = \frac{1}{\Delta}\binom{1}{a + \Delta}, \tag{A.6}$$

$$\frac{d^2\boldsymbol{p}}{d\beta^2} = -\frac{2b}{\Delta^3}\binom{1}{a}, \tag{A.7}$$

$$\frac{d^3\boldsymbol{p}}{d\beta^3} = \frac{12b^2}{\Delta^5}\binom{1}{a}, \quad \text{and} \tag{A.8}$$

$$\frac{d^4\boldsymbol{p}}{d\beta^4} = -\frac{120b^3}{\Delta^7}\binom{1}{a}, \tag{A.9}$$

where the fourth-order derivative (A.9) follows by a similar calculation. Combining these yields the fourth-order Taylor expansion (2.12), in Section 2.1.1.

By requiring $x(\beta_0) = x_0$, one can see that Δ^2 is the discriminant of the polynomial which defines the exact solution (2.13). Therefore, Δ vanishes if and only if (2.13) has exactly one

solution, which is to say that $F = 0$ (2.11) undergoes a bifurcation. For this example, this is also equivalent the Jacobian of F being singular.

B Proofs for high-order implicit derivatives of an operator's root

B.1 Proof of the formula for an operator's high-order β-expansion, Theorem 1

We prove formula (2.15) (of Theorem 1) for a the expansion of $\frac{d^l}{d\beta^l}F(\boldsymbol{x}(\beta), \beta)$. The preliminaries for this subsection are provided in Section II.4.

To get a grip on the proof, we calculate $\frac{d^l F_i}{d\beta^l}$ for $l = 1$ and 2 directly from Faà di Bruno's formula Ma [2009], recovering the first two expansion orders (2.8)-(2.9) in Section I.2.1. Using the notation of Section II.4, write $\mathcal{D}$ for $(T+1)$-decompositions of an integer $l > 0$. Recall, a $(T+1)$-decomposition $(s, \boldsymbol{p}, \boldsymbol{m})$ of l to s parts, $1 \leq s \leq l$, is comprised of parts $0 < p_1 < \cdots < p_s \in \mathbb{N}$ and multiplicities $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_s \in \mathbb{N}_0^{T+1}$, satisfying the decomposition equation (4.4),

$$l = |\boldsymbol{m}_1|p_1 + |\boldsymbol{m}_2|p_2 + \cdots + |\boldsymbol{m}_s|p_s . \tag{B.1}$$

We denote the j-th coordinate of the vector $\boldsymbol{m}_k$ by $m_{k,j}$, where $k = 1, \ldots, s$ and $j = 0, \ldots, T$. Write $\boldsymbol{m} := \boldsymbol{m}_1 + \cdots + \boldsymbol{m}_s \in \mathbb{N}_0^{T+1}$ for the total multiplicity (4.5). Considering F_i as a function of $(\beta, \boldsymbol{x}) \in \mathbb{R}^{T+1}$, we index its β-coordinate by zero, and those of $\boldsymbol{x}$ by $1, \ldots, T$. For a multi-index $\boldsymbol{m} \in \mathbb{N}_0^{T+1}$, write m_0 for its zeroth coordinate and $\boldsymbol{m}_+$ for its other T entries, as above. With this, the proof below of Theorem 1 starts with

$$\frac{d^l}{d\beta^l}F_i(\boldsymbol{x}(\beta), \beta) = l! \sum_{(s, \boldsymbol{p}, \boldsymbol{m}) \in \mathcal{D}} \frac{\partial^{|\boldsymbol{m}|}F_i}{\partial\beta^{m_0}\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^s \frac{1}{\boldsymbol{m}_k!} \left[\frac{1}{p_k!}\frac{d^{p_k}\beta}{d\beta^{p_k}}\right]^{m_{k,0}} \left[\frac{1}{p_k!}\frac{d^{p_k}\boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}} \tag{B.10}$$

Set $l = 1$. By the decomposition equation (B.1), there is only one part of size $p_1 = 1$, and multiplicity $|\boldsymbol{m}_1| = 1$. In particular, we must have $\boldsymbol{m}_1 = \boldsymbol{e}_j$, for $\boldsymbol{e}_j$ a standard basis vector. Where, either $\boldsymbol{m}_1 = \boldsymbol{e}_0$ points to the β-coordinates, or $\boldsymbol{m}_1 = \boldsymbol{e}_j$ points to one of the $\boldsymbol{x}$-coordinates, $j = 1, \ldots, T$. Since these are all the $(T+1)$-decompositions for $l = 1$,

$$\frac{dF_i}{d\beta} \stackrel{\text{(B.10)}}{=} \sum_{\substack{\boldsymbol{m}=\boldsymbol{e}_j: \\ j=0,1,\ldots,T}} \frac{\partial F_i}{\partial\beta^{m_0}\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \left[\frac{d\beta}{d\beta}\right]^{m_0} \left[\frac{d\boldsymbol{x}}{d\beta}\right]^{\boldsymbol{m}_+} = \frac{\partial F_i}{\partial\beta} + \sum_{j=1}^T \frac{\partial F_i}{\partial x_j}\frac{dx_j}{d\beta} . \tag{B.2}$$

This recovers the implicit first-order expansion $\frac{d^1 F}{d\beta^1} = D_{\boldsymbol{x}}F[\frac{d\boldsymbol{x}}{d\beta}] + D_\beta F$ (2.8).

For $l = 2$, setting $p_1 = 2$ again necessitates $\boldsymbol{m}_1 = \boldsymbol{e}_j$ (one part of size 2). Since $p_1 > 1$, the term $\left[\frac{1}{p_1!}\frac{d^{p_1}\beta}{d\beta^{p_1}}\right]^{m_{1,0}}$ at (B.10) vanishes unless $m_{1,0} = 0$, and so only decompositions with $\boldsymbol{m}_1 = \boldsymbol{e}_j$ for $j > 0$ contribute. This yields the term $D_x F_i[\frac{d^2\boldsymbol{x}}{d\beta^2}]$ in the implicit second-order expansion (2.9),

$$2! \sum_{j=1}^T \frac{\partial F_i}{\partial x_j} \left[\frac{1}{2!}\frac{d^2 x_j}{d\beta^2}\right] . \tag{B.3}$$

When $p_1 = 1$, the decomposition equation (B.1) implies that there is only $s = 1$ part, of multiplicity $|\boldsymbol{m}_1| = 2$. Since $p_1 = 1$ we do not have the above restriction on the value of $m_{1,0}$, and so

$\boldsymbol{m}_1 = \boldsymbol{e}_j + \boldsymbol{e}_k$, for $0 \le j \le k \le T$. Further sub-dividing to the cases $j = k = 0, j = 0 < k$ and $0 < j \le k$, we obtain

$$2! \frac{\partial^2 F_i}{\partial \beta^2} \frac{1}{2!} [1]^2 + 2! \sum_{k=1}^{T} \frac{\partial^2 F_i}{\partial \beta \partial x_k} \frac{dx_k}{d\beta}$$

$$+ 2! \sum_{0 < j < k \le T} \frac{\partial^2 F_i}{\partial x_j \partial x_k} \frac{dx_j}{d\beta} \frac{dx_k}{d\beta} + 2! \sum_{0 < j = k \le T} \frac{1}{2!} \frac{\partial^2 F_i}{\partial x_j \partial x_k} \frac{dx_j}{d\beta} \frac{dx_k}{d\beta} \quad \text{(B.4)}$$

The first two terms account respectively for $D^2_{\beta,\beta} F_i$ and $2 D^2_{\beta,x} F_i[\frac{d\boldsymbol{x}}{d\beta}]$ at (2.9), while the last line sums up to the remaining term $D^2_{x,x} F_i[\frac{d\boldsymbol{x}}{d\beta}, \frac{d\boldsymbol{x}}{d\beta}] = \sum_{j,k} \frac{\partial^2 F_i}{\partial x_j \partial x_k} \frac{dx_j}{d\beta} \frac{dx_k}{d\beta}$ there. This can either be seen directly, or as a special case of the following lemma, on rearranging summation order.

Lemma 31. *Let f be a function on $\mathbb{N}_0^T$, and $0 < M \in \mathbb{N}$. Then,*

$$\sum_{\boldsymbol{m} \in \mathbb{N}_0^T : |\boldsymbol{m}| = M} \frac{1}{\boldsymbol{m}!} f(\boldsymbol{m}) = \frac{1}{M!} \sum_{1 \le i_1, \dots, i_M \le T} f(\boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_M}) \quad \text{(B.5)}$$

Proof of Lemma 31. We would like to rewrite the sum over $\{\boldsymbol{m} \in \mathbb{N}_0^T : |\boldsymbol{m}| = M\}$ as a sum over all M-tuples $1 \le i_1, \dots, i_M \le T$. Write $\boldsymbol{m} \in \mathbb{N}_0^T$ with $|\boldsymbol{m}| = M$ as a sum

$$\boldsymbol{m} = \boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_M} \quad \text{(B.6)}$$

of standard basis vectors $\boldsymbol{e}_i$, with each index $1 \le i_j \le T$ for $j = 1, \dots, M$. When the indices $i_1, \dots, i_M$ are distinct, then permuting them yields a different M-tuple of indices, without affecting their sum $\boldsymbol{m}$ at (B.6). A single $\boldsymbol{m}$ value then corresponds to $M!$ distinct permutation of $(i_1, \dots, i_M)$. The indices are distinct if and only if no coordinate of $\boldsymbol{m}$ is larger than 1, $m_k \le 1 \ \forall k = 1, \dots, T$. So,

$$\sum_{\substack{\boldsymbol{m} \in \mathbb{N}_0^T : |\boldsymbol{m}| = M, \\ \forall k, \ m_k \le 1}} M! \, f(\boldsymbol{m}) = \sum_{\substack{1 \le i_1, \dots, i_M \le T : \\ i_1, \dots, i_M \text{ are distinct}}} f(\boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_M}) \quad \text{(B.7)}$$

When exactly two indices i_{j_1} and i_{j_2} for $j_1 \ne j_2$ are the same $i_{j_1} = i_{j_2}$, with all the others distinct, then swapping i_{j_1} with i_{j_2} leaves the M-tuple $(i_1, \dots, i_M)$ unchanged. Hence, there are only $M!/2!$ distinct M-tuples corresponding to the sum $\boldsymbol{m}$ (B.6). Since $i_{j_1} = i_{j_2}$ are the only identical indices, then $m_k = 2$ for $k := i_{j_1}$ and otherwise $m_k \le 1$. Proceeding in this manner, one has in general

$$\sum_{\boldsymbol{m} \in \mathbb{N}_0^T : |\boldsymbol{m}| = M} \frac{M!}{\boldsymbol{m}!} f(\boldsymbol{m}) = \sum_{1 \le i_1, \dots, i_M \le T} f(\boldsymbol{e}_{i_1} + \cdots + \boldsymbol{e}_{i_M}) \quad \text{(B.8)}$$

$\square$

Lemma 31 will allow us to exchange summations over high-dimensional decompositions $(s, \boldsymbol{p}, \boldsymbol{m})$ with summations over integer partitions and multi-linear differentials of F_i.

Proof of Theorem 1. The i-th coordinate of F can be written as a composition

$$\beta \in \mathbb{R} \longrightarrow \big(\beta, \boldsymbol{x}(\beta)\big) \in \mathbb{R}^{T+1} \longrightarrow F_i\big(\boldsymbol{x}(\beta), \beta\big) \in \mathbb{R} \ . \quad \text{(B.9)}$$

We shall fully expand the l-th order derivative $\frac{d^l F_i}{d\beta^l}$ of $F_i\big(\boldsymbol{x}(\beta),\beta\big)$ with respect to β, using the multivariate Faà di Bruno's formula Ma [2009] (Theorem 7 in Section II.4). By the formula,

$$\frac{d^l F_i}{d\beta^l} \overset{(4.6)}{=} l! \sum_{(s,\boldsymbol{p},\boldsymbol{m})\in\mathcal{D}} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial\beta^{m_0}\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^{s} \frac{1}{\boldsymbol{m}_k!}\left[\frac{1}{p_k!}\frac{d^{p_k}(\beta,\boldsymbol{x})}{d\beta^{p_k}}\right]^{\boldsymbol{m}_k}$$

$$\overset{(4.3)}{=} l! \sum_{(s,\boldsymbol{p},\boldsymbol{m})\in\mathcal{D}} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial\beta^{m_0}\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^{s} \frac{1}{\boldsymbol{m}_k!}\left[\frac{1}{p_k!}\frac{d^{p_k}\beta}{d\beta^{p_k}}\right]^{m_{k,0}}\left[\frac{1}{p_k!}\frac{d^{p_k}\boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}} \quad \text{(B.10)}$$

where $\boldsymbol{m} := \boldsymbol{m}_1 + \cdots + \boldsymbol{m}_s \in \mathbb{N}_0^{T+1}$ is the total multiplicity (4.5), and the last equality since we index the β-coordinate of $\mathbb{R}^{T+1}$ by 0, and those of $\boldsymbol{x}$ by $1,\ldots,T$; see (B.1) ff.

With the notation preceding Theorem 7, the parts p_k of a decomposition are positive integers, strictly increasing in $k = 1,\ldots,s$. Hence, only the first part can be of size 1, $p_1 \geq 1$, while the others are of size 2 at least: $p_k > 2$ for $k > 1$. Note that the same differential operator $\frac{d^{p_k}}{d\beta^{p_k}}$ appears for all the $T+1$ coordinates of $\big(\beta,\boldsymbol{x}(\beta)\big)$ at (B.10). At the zeroth entry of the k-th part, the multiplicand $\left[\frac{1}{p_k!}\frac{d^{p_k}\beta}{d\beta^{p_k}}\right]^{m_{k,0}}$ vanishes unless either $m_{k,0} = 0$ or $p_k = 1$. Since p_k can be 1 only for $k = 1$, the latter is to say $p_1 = 1$. In particular, for the k-th summand to contribute, $m_{k,0}$ must vanish when $k > 1$.

To simplify notation, write b instead of $m_{1,0}$. When a summand does not vanish, b is the number of times F_i is differentiated with respect to β (since $m_{k,0} = 0$ except perhaps for $k = 1$). With these observations, decompositions that contribute to (B.10) are as follows:

1. Decompositions with $p_1 > 1$ and $b = 0$.

2. Decompositions with $p_1 = 1$ and any $b = 0,\ldots,|\boldsymbol{m}_1|$.

To proceed, write $M_1 := |\boldsymbol{m}_1|,\ldots,M_s := |\boldsymbol{m}_s|$. The decomposition equation (B.1) can then be read as a partition of the integer l, to M_1 sets of sizes p_1 up to M_s sets of size p_s (and no sets of other sizes),

$$l = M_1 \cdot p_1 + \cdots + M_s \cdot p_s\,. \quad \text{(B.11)}$$

With this, the summation over $(s,\boldsymbol{p},\boldsymbol{m}) \in \mathcal{D}$ in (B.10) can be rewritten as a sum over integer partitions of l. Given a partition (B.11) to parts $p_1,\ldots,p_s$ of respective multiplicities $M_1,\ldots,M_s$, we need to sum over all the integral vectors $\boldsymbol{m}_1,\ldots,\boldsymbol{m}_s \in \mathbb{N}_0^{T+1}$ with $|\boldsymbol{m}_1| = M_1,\ldots,|\boldsymbol{m}_s| = M_s$. This yields,

$$\frac{d^l F_i}{d\beta^l} \overset{(B.10)}{=} l! \sum_{(s,\boldsymbol{p},\boldsymbol{m})\in\mathcal{D}} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial\beta^{m_0}\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^{s} \frac{1}{\boldsymbol{m}_k!}\left[\frac{1}{p_k!}\frac{d^{p_k}\beta}{d\beta^{p_k}}\right]^{m_{k,0}}\left[\frac{1}{p_k!}\frac{d^{p_k}\boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}$$

$$= l! \sum_{\substack{\text{partitions} \\ \text{(B.11) of } l}} \sum_{\boldsymbol{m}_1:|\boldsymbol{m}_1|=M_1} \cdots \sum_{\boldsymbol{m}_s:|\boldsymbol{m}_s|=M_s} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial\beta^b\,\partial\boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^{s} \frac{1}{\boldsymbol{m}_k!\,p_k!^{|\boldsymbol{m}_k|}}\left[\frac{d^{p_k}\beta}{d\beta^{p_k}}\right]^{m_{k,0}}\left[\frac{d^{p_k}\boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}$$

$$\text{(B.12)}$$

where in the last line we replaced $m_{1,0}$ by b, using the fact that $m_{k,0}$ must vanish for $k > 1$, and so $m_0 = m_{1,0} + m_{2,0} + \cdots + m_{s,0} = b$.

The summation over $\boldsymbol{m}_1$ can be broken to two, isolating its zeroth component b from the rest, as $|\boldsymbol{m}_1| = b + |\boldsymbol{m}_{1+}|$ by definition. Recall that decompositions with $b > 0$ contribute only when $p_1 = 1$, as in 2, so

$$\sum_{\boldsymbol{m}_1:|\boldsymbol{m}_1|=M_1} \cdots = \sum_{b=0}^{M_1\cdot\delta(p_1=1)} \sum_{\boldsymbol{m}_{1+}:|\boldsymbol{m}_{1+}|=M_1-b} \cdots \quad \text{(B.13)}$$

Other than $m_{1,0} =: b$, $m_{k,0}$ always vanishes. So, we lose nothing by replacing our $(T+1)$-dimensional multiplicity vectors $\boldsymbol{m}_k$ with smaller ones $\boldsymbol{m}_{k+} \in \mathbb{N}_0^T$. So,

$$\frac{d^l F_i}{d\beta^l} \overset{\text{(B.12)}}{\underset{\text{(B.13)}}{=}} \sum_{\text{partitions of } l} \sum_{b=0}^{M_1 \cdot \delta(p_1=1)} \frac{l!}{b! \, (p_1!)^{m_1} \cdots (p_s!)^{m_s}} \sum_{\boldsymbol{m}_{1+}: |\boldsymbol{m}_{1+}|=M_1-b} \sum_{\boldsymbol{m}_{2+}: |\boldsymbol{m}_{2+}|=M_2} \cdots$$
$$\cdots \sum_{\boldsymbol{m}_{s+}: |\boldsymbol{m}_{s+}|=M_s} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial \beta^b \, \partial x^{\boldsymbol{m}_+}} \prod_{k=1}^s \frac{1}{\boldsymbol{m}_{k+}!} \left[\frac{d^{p_k} \boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}} \tag{B.14}$$

where the $^1/b!$ coefficient in the first line is due to $\boldsymbol{m}_1! = b! \cdot \boldsymbol{m}_{1+}!$, by definition (4.3).

Next, for $s > 1$, the derivative term $\partial^{|\boldsymbol{m}|} F_i$ in the last line of (B.14) can be written as

$$\frac{\partial^{|\boldsymbol{m}|} F_i}{\partial \beta^b \, \partial \boldsymbol{x}^{\boldsymbol{m}_+}} = \frac{\partial^{M_s}}{\partial \boldsymbol{x}^{\boldsymbol{m}_{s+}}} \frac{\partial^{|\boldsymbol{m}|-M_s} F_i}{\partial \beta^b \, \partial \boldsymbol{x}^{\boldsymbol{m}_{1+}} \cdots \partial \boldsymbol{x}^{\boldsymbol{m}_{(s-1)+}}} =: \frac{\partial^{M_s}}{\partial \boldsymbol{x}^{\boldsymbol{m}_{s+}}} G \, . \tag{B.15}$$

While G itself does *not* depend on the multi-index $\boldsymbol{m}_{s+}$, we consider $\frac{\partial^{M_s}}{\partial \boldsymbol{x}^{\boldsymbol{m}_{s+}}} G$ as a function of $\boldsymbol{m}_{s+}$. This allows us to invoke Lemma 31 on the last summation at (B.14), as follows:

$$\sum_{\boldsymbol{m}_{s+}: |\boldsymbol{m}_{s+}|=M_s} \frac{\partial^{|\boldsymbol{m}|} F_i}{\partial \beta^b \, \partial \boldsymbol{x}^{\boldsymbol{m}_+}} \prod_{k=1}^s \frac{1}{\boldsymbol{m}_{k+}!} \left[\frac{d^{p_k} \boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}$$
$$\overset{\text{(B.15)}}{=} \left(\prod_{k=1}^{s-1} \frac{1}{\boldsymbol{m}_{k+}!} \left[\frac{d^{p_k} \boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}\right) \cdot \sum_{\boldsymbol{m}_{s+}: |\boldsymbol{m}_{s+}|=M_s} \frac{1}{\boldsymbol{m}_{s+}!} \cdot \frac{\partial^{M_s} G}{\partial \boldsymbol{x}^{\boldsymbol{m}_{s+}}} \left[\frac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}\right]^{\boldsymbol{m}_{s+}}$$
$$\overset{\text{Lemma 31}}{=} \left(\prod_{k=1}^{s-1} \frac{1}{\boldsymbol{m}_{k+}!} \left[\frac{d^{p_k} \boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}\right) \cdot \frac{1}{M_S!} \sum_{1 \leq i_1, \ldots, i_{M_s} \leq T} \frac{\partial^{M_s} G}{\partial x_{i_1} \cdots \partial x_{i_{M_s}}} \cdot \frac{d^{p_s} x_{i_1}}{d\beta^{p_s}} \cdots \frac{d^{p_s} x_{i_{M_s}}}{d\beta^{p_s}}$$
$$\overset{\text{(2.6)}}{=} \frac{1}{M_S!} \left(\prod_{k=1}^{s-1} \frac{1}{\boldsymbol{m}_{k+}!} \left[\frac{d^{p_k} \boldsymbol{x}}{d\beta^{p_k}}\right]^{\boldsymbol{m}_{k+}}\right) \cdot D^{M_s}_{\underbrace{\boldsymbol{x}, \ldots, \boldsymbol{x}}_{M_s \text{ times}}} G\Big[\underbrace{\tfrac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}, \ldots, \tfrac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}}_{M_s \text{ times}}\Big] \tag{B.16}$$

where the last line is by the definition of a multivariate derivative tensor, (2.6) in Section I.2.1.

The exact same manipulations as in (B.16) can be applied to all the other partition parts $k = 1, \ldots, s-1$, with one caveat. For $k = 1$, $|\boldsymbol{m}_{1+}|$ is $M_1 - b$ rather than M_1, while the remaining b degrees are consumed by the derivative with respect to β, as can be seen from (B.15). Thus, the coefficient at the last line of (B.16) ends being $((M_1 - b)! \cdot M_2! \cdots M_s!)^{-1}$, and the differentiation corresponding to $k = 1$ is b-times with respect to β, and only $(M_1 - b)$ times with respect to $\boldsymbol{x}$. In particular, it is then an $(M_1 - b)$-multilinear form, involving $\frac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}}$ only $(M_1 - b)$ times. Gathering these back to (B.14) for all parts $k = 1, \ldots, s$ completes the proof, yielding (2.15). $\qquad \square$

B.2 Proof of the formula for the derivative's Jacobian, Proposition 27

Proof of Proposition 27. To avoid clutter, we write $D_{\boldsymbol{x}} F^{-1}$ in this proof for the inverse of the Jacobian matrix $D_{\boldsymbol{x}} F$. Rewrite formula (2.17) (Theorem 3) compactly as,

$$D_{\boldsymbol{x}} F \, \frac{d^l \boldsymbol{x}}{d\beta^l} = -S \tag{B.17}$$

where for short,

$$S := \sum_{\substack{\text{non-trivial} \\ \text{partitions}}} \sum_{b=0}^{m_1 \cdot \delta(p_1=1)} C \cdot D^m_{\beta^b, \boldsymbol{x}^{m-b}} F\Big[\Big(\tfrac{d^{p_1} \boldsymbol{x}}{d\beta^{p_1}}\Big)_{\times (m_1-b)}, \Big(\tfrac{d^{p_2} \boldsymbol{x}}{d\beta^{p_2}}\Big)_{\times m_2}, \ldots, \Big(\tfrac{d^{p_s} \boldsymbol{x}}{d\beta^{p_s}}\Big)_{\times m_s}\Big], \tag{B.18}$$

and C stands for the coefficient $\frac{l!}{b!(m_1-b)!m_2!\cdots m_s!\cdot(p_1!)^{m_1}\cdots(p_s!)^{m_s}}$ at (2.17). Differentiating both sides of (B.17) with respect to the coordinates $\boldsymbol{x}$,

$$D^2_{\boldsymbol{x},\boldsymbol{x}}F\,\tfrac{d^l\boldsymbol{x}}{d\beta^l} + D_{\boldsymbol{x}}F\;D_{\boldsymbol{x}}\tfrac{d^l\boldsymbol{x}}{d\beta^l} = -D_{\boldsymbol{x}}S \tag{B.19}$$

That is,

$$D_{\boldsymbol{x}}\tfrac{d^l\boldsymbol{x}}{d\beta^l} = -D_{\boldsymbol{x}}F^{-1}\left(D^2_{\boldsymbol{x},\boldsymbol{x}}F\right)\tfrac{d^l\boldsymbol{x}}{d\beta^l} - D_{\boldsymbol{x}}F^{-1}D_{\boldsymbol{x}}S \tag{B.20}$$

To complete the proof, it suffices to calculate $D_{\boldsymbol{x}}S$.

The differentiation of a single addend in S (B.18) is a sum of 2-tensors (matrices). Each addend in this sum involves extra $\boldsymbol{x}$-differentiations: once of F itself, and once for each of its m arguments. Differentiating a single addend,

$$
\begin{aligned}
&D_{\boldsymbol{x}}\left(D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\right)\\
&\quad= D^m_{\beta^b,\boldsymbol{x}^{m-b+1}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[D_{\boldsymbol{x}}\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\underbrace{\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\cdots,\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}}_{(m_1-b-1)\text{ times}},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\underbrace{\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}}_{\text{once}},D_{\boldsymbol{x}}\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\underbrace{\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\cdots,\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}}_{(m_1-b-2)\text{ times}},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]+\cdots\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\underbrace{\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\cdots,\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}}_{(m_1-b-1)\text{ times}},D_{\boldsymbol{x}}\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad+\cdots\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,D_{\boldsymbol{x}}\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\underbrace{\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\cdots,\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}}_{m_s-1\text{ times}}\right]\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\underbrace{\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}}_{\text{once}},D_{\boldsymbol{x}}\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\underbrace{\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\cdots,\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}}_{m_s-2\text{ times}}\right]+\cdots\\
&\quad+ D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\underbrace{\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\cdots,\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}}_{m_s-1\text{ times}},D_{\boldsymbol{x}}\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right]
\end{aligned}
\tag{B.21}
$$

Since derivative tensors of high-order are symmetric (e.g., [Aguilar, 2021, Section 10.3]), permuting its arguments has no effect. So, the above simplifies to

$$
\begin{aligned}
&D_{\boldsymbol{x}}\left(D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\right)\\
&\quad= D^m_{\beta^b,\boldsymbol{x}^{m-b+1}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\quad+ (m_1-b)\cdot D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[D_{\boldsymbol{x}}\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}},\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b-1)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\quad+ m_2\cdot D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},D_{\boldsymbol{x}}\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times(m_2-1)},\cdots,\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times m_s}\right]\\
&\qquad\qquad\qquad\qquad\qquad+\cdots\\
&\quad+ m_s\cdot D^m_{\beta^b,\boldsymbol{x}^{m-b}}F\left[\left(\tfrac{d^{p_1}\boldsymbol{x}}{d\beta^{p_1}}\right)_{\times(m_1-b)},\left(\tfrac{d^{p_2}\boldsymbol{x}}{d\beta^{p_2}}\right)_{\times m_2},\cdots,D_{\boldsymbol{x}}\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}},\left(\tfrac{d^{p_s}\boldsymbol{x}}{d\beta^{p_s}}\right)_{\times(m_s-1)}\right]
\end{aligned}
\tag{B.22}
$$

Combining the latter (B.22) with the definition (B.18) of S and Equation (B.20) yields the required result (7.11). $\qquad\square$

C Derivations of high-order derivatives of the Blahut-Arimoto operator

In this section, we calculated derivatives of the Blahut-Arimoto operators, mainly those presented in Section 5.

C.1 Proof of Proposition 9, formula for the encoder's repeated marginal derivatives

To prove Proposition 9, we rewrite it in an equivalent form which is more convenient for proofs by induction. Rather than writing high-order derivatives in multi-index notation, one could write it as a sequence of differentiations. Setting $M = 2$ for example, a third-order derivative represented by $\boldsymbol{\alpha}_+ = (2, 1)$ can be written equivalently as

$$\frac{\partial^3}{\partial \boldsymbol{r}^{(2,1)}} q(\hat{x}|x) = \frac{\partial^3}{\partial r(\hat{x}_1)\partial r(\hat{x}_1)\partial r(\hat{x}_2)} q(\hat{x}|x) . \tag{C.1}$$

While the left-hand side of (C.1) is understood (4.3) as an application of the differential operator $\left(\frac{\partial}{\partial r(\hat{x}_1)}\right)^2 \cdot \left(\frac{\partial}{\partial r(\hat{x}_2)}\right)$, its right-hand side can be considered as a sequence of differentiations: first differentiate with respect to $r(\hat{x}_2)$, and then twice with respect to $r(\hat{x}_1)$.

Write $\langle \cdot, \cdot \rangle$ for the usual scalar product on $\mathbb{R}^M$, $\boldsymbol{e}_{\hat{x}_i}$ for the i-th standard basis vector. Then, $\langle \boldsymbol{\alpha}_+, \boldsymbol{e}_{\hat{x}_i} \rangle = \alpha_i$ is the number differentiations with respect to the i-th coordinate $r(\hat{x}_i)$, and $|\boldsymbol{\alpha}_+|$ the number of differentiations in total. When differentiating with respect to $r(\hat{x}_{i_1}), r(\hat{x}_{i_2})$ up to $r(\hat{x}_{i_k})$, the total number $\alpha_{\hat{x}'}$ of differentiations with respect to a particular coordinate $\hat{x}'$ can be written as $\sum_{j=1}^{k} \delta_{\hat{x}', \hat{x}_{i_j}} = \delta_{\hat{x}', \hat{x}_{i_1}} + \cdots + \delta_{\hat{x}', \hat{x}_{i_k}}$. Thus, Proposition 9 is equivalent to the following.

Proposition 32. *For $k > 0$, the repeated encoder* (2.18) *derivative with respect to $\boldsymbol{r}$ is,*

$$\frac{\partial^k}{\partial r(\hat{x}_{i_1}) \cdots \partial r(\hat{x}_{i_k})} q(\hat{x}|x) = \frac{(-1)^{k-1}(k-1)! \, e^{-\beta \sum_{j=1}^{k} d(x, \hat{x}_{i_j})}}{Z^k(x, \beta)} \cdot \left[\sum_{j=1}^{k} \delta_{\hat{x}, \hat{x}_{i_j}} - k \cdot q(\hat{x}|x) \right] \tag{C.2}$$

where $1 \leq i_1, \ldots, i_k \leq M$ need not be distinct.

Proof of Proposition 32. We prove (C.2) by induction on $k > 0$. First, note that

$$\frac{\partial}{\partial r(\hat{x})} Z(x, \beta) = \frac{\partial}{\partial r(\hat{x})} \sum_{\hat{x}'} r(\hat{x}') e^{-\beta d(x, \hat{x}')} = e^{-\beta d(x, \hat{x})} \tag{C.3}$$

So, for $k > 0$ we have

$$\frac{\partial}{\partial r(\hat{x})} \frac{1}{Z^k(x, \beta)} \overset{\text{(C.3)}}{=} -\frac{k e^{-\beta d(x, \hat{x})}}{Z^{k+1}(x, \beta)} \tag{C.4}$$

Hence,

$$\frac{\partial}{\partial r(\hat{x}')} \left(\frac{r(\hat{x})}{Z^k(x, \beta)} \right) = \frac{\delta_{\hat{x}, \hat{x}'}}{Z^k(x, \beta)} - \frac{k e^{-\beta d(x, \hat{x}')}}{Z^{k+1}(x, \beta)} \cdot r(\hat{x}) \tag{C.5}$$

Thus, for the first derivative of the encoder (2.18) we have

$$\frac{\partial}{\partial r(\hat{x}_{i_1})} q(\hat{x}|x) \overset{(2.18)}{=} e^{-\beta d(x,\hat{x})} \cdot \frac{\partial}{\partial r(\hat{x}_{i_1})} \left(\frac{r(\hat{x})}{Z(x,\beta)} \right)$$

$$\overset{(C.5)}{=} e^{-\beta d(x,\hat{x})} \cdot \left\{ \frac{\delta_{\hat{x},\hat{x}_{i_1}}}{Z(x,\beta)} - \frac{e^{-\beta d(x,\hat{x}_{i_1})}}{Z^2(x,\beta)} \cdot r(\hat{x}) \right\}$$

$$= \frac{e^{-\beta d(x,\hat{x}_{i_1})}}{Z(x,\beta)} \cdot \left\{ e^{-\beta d(x,\hat{x})} \cdot \frac{\delta_{\hat{x},\hat{x}_{i_1}}}{e^{-\beta d(x,\hat{x}_{i_1})}} - \frac{e^{-\beta d(x,\hat{x})}}{Z(x,\beta)} \cdot r(\hat{x}) \right\}$$

$$\overset{(2.18)}{=} \frac{e^{-\beta d(x,\hat{x}_{i_1})}}{Z(x,\beta)} \cdot \left\{ \delta_{\hat{x},\hat{x}_{i_1}} - q(\hat{x}|x) \right\} \quad (C.6)$$

This is the induction basis $k = 1$ for $(C.2)$.

Next, assume that $(C.2)$ holds for any derivative up to order k. Then, for $k + 1$ we have,

$$\frac{\partial}{\partial r(\hat{x}_{i_{k+1}})} \frac{\partial^k}{\partial r(\hat{x}_{i_1}) \cdots \partial r(\hat{x}_{i_k})} q(\hat{x}|x)$$

$$\overset{(C.2)}{=} \frac{\partial}{\partial r(\hat{x}_{i_{k+1}})} \left[\frac{(-1)^{k-1}(k-1)! \, e^{-\beta \sum_{j=1}^k d(x,\hat{x}_{i_j})}}{Z^k(x,\beta)} \cdot \left\{ \sum_{j=1}^k \delta_{\hat{x},\hat{x}_{i_j}} - k \cdot q(\hat{x}|x) \right\} \right]$$

$$\overset{(C.4)}{\underset{(C.6)}{=}} \frac{(-1)^k \, k! \, e^{-\beta \sum_{j=1}^k d(x,\hat{x}_{i_j})}}{Z^{k+1}(x,\beta)} \cdot e^{-\beta d(x,\hat{x}_{i_{k+1}})} \cdot \left\{ \sum_{j=1}^k \delta_{\hat{x},\hat{x}_{i_j}} - k \cdot q(\hat{x}|x) \right\}$$

$$+ \frac{(-1)^{k-1}(k-1)! \, e^{-\beta \sum_{j=1}^k d(x,\hat{x}_{i_j})}}{Z^k(x,\beta)} \cdot (-k) \cdot \frac{e^{-\beta d(x,\hat{x}_{i_{k+1}})}}{Z(x,\beta)} \cdot \left\{ \delta_{\hat{x},\hat{x}_{i_{k+1}}} - q(\hat{x}|x) \right\}$$

$$= \frac{(-1)^k \, k! \, e^{-\beta \sum_{j=1}^{k+1} d(x,\hat{x}_{i_j})}}{Z^{k+1}(x,\beta)} \cdot \left\{ \sum_{j=1}^{k+1} \delta_{\hat{x},\hat{x}_{i_j}} - (k+1) \cdot q(\hat{x}|x) \right\} \quad (C.7)$$

This completes the proof of the induction step. $\qquad \square$

C.2 Proof of Proposition 12, formula for the encoder's repeated β-derivatives

Proof of Proposition 12. Note that,

$$\frac{\partial}{\partial \beta} e^{-\beta d(x,\hat{x})} = -d(x,\hat{x}) e^{-\beta d(x,\hat{x})}$$

$$\implies \frac{\partial}{\partial \beta} Z(x,\beta) = \frac{\partial}{\partial \beta} \sum_{\hat{x}'} r(\hat{x}') e^{-\beta d(x,\hat{x}')} = -\sum_{\hat{x}'} d(x,\hat{x}') r(\hat{x}') e^{-\beta d(x,\hat{x}')} \quad (C.8)$$

So,

$$\frac{\partial}{\partial \beta} \frac{1}{Z(x,\beta)} \overset{(C.8)}{=} \frac{1}{Z(x,\beta)} \cdot \sum_{\hat{x}'} d(x,\hat{x}') \frac{r(\hat{x}') e^{-\beta d(x,\hat{x}')}}{Z(x,\beta)} \overset{(2.18)}{=} \frac{1}{Z(x,\beta)} \cdot \sum_{\hat{x}'} q(\hat{x}'|x) d(x,\hat{x}') \quad (C.9)$$

Thus, for the encoder's first-order β-derivative,

$$\frac{\partial q(\hat{x}|x)}{\partial \beta} \overset{(2.18)}{=} r(\hat{x}) \cdot \frac{\partial}{\partial \beta} \frac{e^{-\beta d(x,\hat{x})}}{Z(x,\beta)} \overset{(C.8)}{\underset{(C.9)}{=}} r(\hat{x}) \cdot \left[\frac{e^{-\beta d(x,\hat{x})}}{Z(x,\beta)} \cdot \sum_{\hat{x}'} q(\hat{x}'|x) d(x,\hat{x}') - d(x,\hat{x}) \cdot \frac{e^{-\beta d(x,\hat{x})}}{Z(x,\beta)} \right]$$

$$\overset{(2.18)}{=} q(\hat{x}|x) \cdot \sum_{\hat{x}'} \left(q(\hat{x}'|x) - \delta_{\hat{x},\hat{x}'} \right) d(x,\hat{x}') = q(\hat{x}|x) \cdot \left(\mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}') \right] - d(x,\hat{x}) \right) \quad \text{(C.10)}$$

where we have denoted $\mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right] := \sum_{\hat{x}'} q(\hat{x}'|x) d(x,\hat{x}')^k$, for $k > 0$. Writing x_0 for $d(x,\hat{x})$ and x_k for $\mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right]$ when $k > 0$, this proves the first-order version of formula (5.11), with $P_1(x_0, x_1) := x_1 - x_0$ (2.25).

Unlike the encoder $\boldsymbol{q}$, the distortion $d(x,\hat{x})$ does *not* depend on β. So, for $k > 0$,

$$\frac{\partial}{\partial \beta} \mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right] = \sum_{\hat{x}'} d(x,\hat{x}')^k \frac{\partial q(\hat{x}'|x)}{\partial \beta}$$

$$\overset{(C.10)}{=} \sum_{\hat{x}'} d(x,\hat{x}')^k \, q(\hat{x}'|x) \cdot \sum_{\hat{x}''} \left(q(\hat{x}''|x) - \delta_{\hat{x}',\hat{x}''} \right) d(x,\hat{x}'')$$

$$= \sum_{\hat{x}',\hat{x}''} d(x,\hat{x}')^k \, d(x,\hat{x}'') q(\hat{x}'|x) \left(q(\hat{x}''|x) - \delta_{\hat{x}',\hat{x}''} \right)$$

$$= \mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}') \right] \cdot \mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right] - \mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^{k+1} \right] \quad \text{(C.11)}$$

That is, when writing $x_k := \mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}')^k \right]$ for $k > 0$, the x_k's satisfy the relations $\bar{d}x_k = x_1 \cdot x_k - x_{k+1}$ for $k > 0$ and $\bar{d}x_0 = 0$ (2.22), where $\bar{d}$ is written in place of $\partial/\partial\beta$ to emphasize the algebraic properties of this definition.

Next, assuming that the derivative formula (5.11) holds for k, we prove it for $k+1$. With x_0 and x_k for $k > 0$ as before,

$$\frac{\partial^{k+1} q(\hat{x}|x)}{\partial \beta^{k+1}} = \frac{\partial}{\partial \beta} \frac{\partial^k q(\hat{x}|x)}{\partial \beta^k} \overset{(5.11)}{=} \frac{\partial}{\partial \beta} \left[q(\hat{x}|x) \cdot P_k \left(x_0, x_1, \ldots, x_k \right) \right]$$

$$= \frac{\partial q(\hat{x}|x)}{\partial \beta} \cdot P_k \left(x_0, x_1, \ldots, x_k \right) + q(\hat{x}|x) \cdot \tfrac{\partial}{\partial \beta} P_k \left(x_0, x_1, \ldots, x_k \right)$$

$$\overset{(C.10)}{=} q(\hat{x}|x) \cdot (x_1 - x_0) \cdot P_k \left(x_0, x_1, \ldots, x_k \right) + q(\hat{x}|x) \cdot \bar{d} P_k \left(x_0, x_1, \ldots, x_k \right)$$

$$= q(\hat{x}|x) \cdot \left((x_1 - x_0) \cdot P_k + \bar{d} P_k \right) \quad \text{(C.12)}$$

Where we have replaced $\partial/\partial\beta$ by $\bar{d}$, as before. So, setting $P_{k+1} := (x_1 - x_0) \cdot P_k + \bar{d} P_k$ (2.24) completes the proof. $\qquad \square$

C.3   Proof of Corollary 13, for the partial β-derivative of BA_β

Proof of Corollary 13. The encoder's first partial β-derivative is

$$\frac{\partial q(\hat{x}|x)}{\partial \beta} \overset{(5.11)}{\underset{(2.25)}{=}} q(\hat{x}|x) \cdot \left(\mathbb{E}_{q(\hat{x}'|x)} \left[d(x,\hat{x}') \right] - d(x,\hat{x}) \right) \quad \text{(C.13)}$$

Plugging this back into formula (5.4) for the derivative of BA_β,

$$\frac{\partial BA_\beta \left[r \right](\hat{x})}{\partial \beta} \overset{(5.4)}{=} \sum_x p_X(x) \frac{\partial q(\hat{x}|x)}{\partial \beta} \overset{(C.13)}{=} \sum_x p_X(x) \left[\sum_{\hat{x}'} q(\hat{x}|x) q(\hat{x}'|x) d(x,\hat{x}') - q(\hat{x}|x) d(x,\hat{x}) \right]$$

$$= \mathbb{E}_{q(\hat{x}'|x) p_X(x)} \left[q(\hat{x}|x) d(x,\hat{x}') \right] - \mathbb{E}_{p_X(x)} \left[q(\hat{x}|x) d(x,\hat{x}) \right] \quad \text{(C.14)}$$

Since the identity operator does not depend on β, this yields the result. $\qquad\square$

C.4 Proof of Proposition 15, formula for mixed high-order encoder derivatives

Proof of Proposition 15. Let $\boldsymbol{\alpha} = (\alpha_0, \boldsymbol{\alpha}_+) \in \mathbb{N}_0^{M+1}$ with $\boldsymbol{\alpha}_+ \neq 0$ be given, and an input marginal $\boldsymbol{r}$ outside the simplex boundary, $\forall \hat{x} \; r(\hat{x}) > 0$. Let $\boldsymbol{q}$ be the encoder defined by $\boldsymbol{r}$ (2.18). For a fixed x coordinate, consider $d(x, \hat{x})$ and $q(\hat{x}|x)$ as $\hat{x}$-indexed vectors. Thus, by the multivariate vector-power notation in Section II.4,

$$\frac{e^{-\beta d(x,\hat{x}')}}{Z(x,\beta)} \overset{(2.18)}{=} \frac{q(\hat{x}'|x)}{r(\hat{x}')} \quad \Longrightarrow \quad \left(\frac{e^{-\beta d(x,\hat{x})}}{Z(x,\beta)}\right)^{\boldsymbol{\alpha}_+} \overset{(4.3)}{=} \frac{e^{-\beta\langle \boldsymbol{\alpha}_+, d(x,\hat{x})\rangle}}{Z^{|\boldsymbol{\alpha}_+|}(x,\beta)} = \frac{q(\hat{x}|x)^{\boldsymbol{\alpha}_+}}{\boldsymbol{r}^{\boldsymbol{\alpha}_+}} \tag{C.15}$$

Hence, using Proposition 9,

$$\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial\beta^{\alpha_0}\,\partial\boldsymbol{r}^{\boldsymbol{\alpha}_+}} q(\hat{x}'|x)$$

$$\overset{(5.5)}{=} \frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}} \left\{ \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!\, e^{-\beta\langle \boldsymbol{\alpha}_+, d(x,\hat{x})\rangle}}{Z^{|\boldsymbol{\alpha}_+|}(x,\beta)} \cdot \left[\langle \boldsymbol{\alpha}_+, \boldsymbol{e}_{\hat{x}'}\rangle - |\boldsymbol{\alpha}_+| \cdot q(\hat{x}'|x) \right] \right\}$$

$$\overset{(C.15)}{=} \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!}{\boldsymbol{r}^{\boldsymbol{\alpha}_+}} \cdot \frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}} \left\{ q(\hat{x}|x)^{\boldsymbol{\alpha}_+} \left[\alpha_{\hat{x}'} - |\boldsymbol{\alpha}_+| \cdot q(\hat{x}'|x) \right] \right\} \tag{C.16}$$

Where, $q(\hat{x}|x)^{\boldsymbol{\alpha}_+}$ in the last line is considered as an $\hat{x}$-indexed vector for x fixed, $\boldsymbol{\alpha}_+ \in \mathbb{N}_0^M$.

To proceed, we need a generalization of the Leibniz rule (4.1) to multiple factors $f_1, \ldots, f_m$. It is a direct exercise by induction Thaheem and Laradji [2003] to see that

$$(f_1 f_2 \cdots f_m)^{(n)} = \sum_{|\boldsymbol{k}|=n} \frac{n!}{\boldsymbol{k}!} \prod_{i=1}^{m} f_i^{(k_i)} \tag{C.17}$$

where the sum is over all m-tuples $\boldsymbol{k} := (k_1, \ldots, k_m)$ of non-negative integers with $|\boldsymbol{k}| = \sum_{i=1}^{m} k_i = n$, and $n!/\boldsymbol{k}!$ is the multinomial coefficient,

$$\frac{n!}{\boldsymbol{k}!} = \frac{n!}{k_1!\,k_2!\,\cdots\,k_m!} := \binom{n}{k_1, k_2, \ldots, k_m}. \tag{C.18}$$

Next, break the derivative of a product $\frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}} q(\hat{x}|x)^{\boldsymbol{\alpha}_+}$ (C.16) to a product of derivatives,

$$\frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}} q(\hat{x}|x)^{\boldsymbol{\alpha}_+} \overset{(4.3)}{=} \frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}} \left(\prod_{i=1}^{M} q(\hat{x}_i|x)^{\alpha_i}\right) \overset{(C.17)}{=} \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0} \frac{\alpha_0!}{\boldsymbol{k}!} \prod_{i=1}^{M} \frac{\partial^{k_i}}{\partial\beta^{k_i}}\left(q(\hat{x}_i|x)^{\alpha_i}\right) \tag{C.19}$$

where we have set $n := \alpha_0$ and $m := M$ at (C.17), so that $\boldsymbol{k} \in \mathbb{N}_0^M$. Note that the inner multiplicand to the right vanishes whenever $\alpha_i = 0$ but $k_i > 0$.

To calculate a multiplicand $\frac{\partial^{k_i}}{\partial\beta^{k_i}} q(\hat{x}_i|x)^{\alpha_i}$ at (C.19), we apply the univariate Faà di Bruno's formula (4.2) to $x \mapsto x^{\alpha_i}$ composed after $\beta \mapsto q(\hat{x}_i|x)$. When $k_i > 0$, we have for $\boldsymbol{t} \in \mathbb{N}_0^{k_i}$

$$\frac{d^{|\boldsymbol{t}|}}{dx^{|\boldsymbol{t}|}} x^{\alpha_i} = \delta_{|\boldsymbol{t}| \leq \alpha_i} \cdot \frac{\alpha_i!}{(\alpha_i - |\boldsymbol{t}|)!} x^{\alpha_i - |\boldsymbol{t}|} \tag{C.20}$$

where $\delta_{|\boldsymbol{t}| \le \alpha_i}$ is the Kronecker delta. So, summing over all integer partitions $\boldsymbol{t} = (t_1, \ldots, t_{k_i})$ of k_i to at most $|\boldsymbol{t}| \le \alpha_i$ subsets, $\sum_{j=1}^{k_i} j \cdot t_j = k_i$, we obtain

$$
\frac{\partial^{k_i}}{\partial \beta^{k_i}} q(\hat{x}_i | x)^{\alpha_i} \stackrel{\substack{(4.2)\\(C.20)}}{=} \sum \frac{k_i!}{\boldsymbol{t}! \cdot (1!^{t_1} \cdots k_i!^{t_{k_i}})} \cdot \frac{\alpha_i!}{(\alpha_i - |\boldsymbol{t}|)!} \, q(\hat{x}_i | x)^{\alpha_i - |\boldsymbol{t}|} \cdot \prod_{j=1}^{k_i} \left(\frac{\partial^j q(\hat{x}_i | x)}{\partial \beta^j} \right)^{t_j}
$$

$$
\stackrel{(5.11)}{=} \sum \frac{k_i!}{\boldsymbol{t}! \cdot (1!^{t_1} \cdots k_i!^{t_{k_i}})} \cdot \frac{\alpha_i!}{(\alpha_i - |\boldsymbol{t}|)!} \, q(\hat{x}_i | x)^{\alpha_i - |\boldsymbol{t}|} \cdot \prod_{j=1}^{k_i} q(\hat{x}_i | x)^{t_j} \cdot P_j(\hat{x}_i, x)^{t_j}
$$

$$
= q(\hat{x}_i | x)^{\alpha_i} \cdot \sum \frac{k_i!}{\boldsymbol{t}!} \cdot \frac{\alpha_i!}{(\alpha_i - |\boldsymbol{t}|)!} \prod_{j=1}^{k_i} \left(\frac{P_j(\hat{x}_i, x)}{j!} \right)^{t_j} \qquad \text{(C.21)}
$$

Where, in the second equality we used formula (5.11) for the encoder's repeated partial β-derivative (Proposition 12), and the last simplifies thanks to $\prod_j q(\hat{x}_i | x)^{t_j} = q(\hat{x}_i | x)^{\sum_j t_j} = q(\hat{x}_i | x)^{|\boldsymbol{t}|}$. Note that if $k_i = 0$, then $\boldsymbol{t} = \boldsymbol{0}$ is the only integer partition of k_i, and so the end result of (C.21) correctly reduces to $q(\hat{x}_i | x)^{\alpha_i}$. Plugging the latter back into (C.19),

$$
\frac{\partial^{\alpha_0}}{\partial \beta^{\alpha_0}} q(\hat{x} | x)^{\boldsymbol{\alpha}_+} \stackrel{(C.19)}{=} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \frac{\alpha_0!}{\boldsymbol{k}!} \prod_{i=1}^M \frac{\partial^{k_i}}{\partial \beta^{k_i}} \left(q(\hat{x}_i | x)^{\alpha_i} \right)
$$

$$
\stackrel{(C.21)}{=} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \frac{\alpha_0!}{\boldsymbol{k}!} \prod_{i=1}^M q(\hat{x}_i | x)^{\alpha_i} \sum_{\substack{\boldsymbol{t}: \, |\boldsymbol{t}| \le \alpha_i, \\ \sum_j j \cdot t_j = k_i}} \frac{k_i!}{\boldsymbol{t}!} \cdot \frac{\alpha_i!}{(\alpha_i - |\boldsymbol{t}|)!} \prod_{j=1}^{k_i} \left(\frac{P_j(\hat{x}_i, x)}{j!} \right)^{t_j}
$$

$$
= \boldsymbol{\alpha}! \, q(\hat{x} | x)^{\boldsymbol{\alpha}_+} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \prod_{i=1}^M \sum_{\substack{\boldsymbol{t}: \, |\boldsymbol{t}| \le \alpha_i, \\ \sum_j j \cdot t_j = k_i}} \frac{1}{\boldsymbol{t}! \, (\alpha_i - |\boldsymbol{t}|)!} \prod_{j=1}^{k_i} \left(\frac{P_j(\hat{x}_i, x)}{j!} \right)^{t_j}
$$

$$
=: \boldsymbol{\alpha}! \, q(\hat{x} | x)^{\boldsymbol{\alpha}_+} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \prod_{i=1}^M G\big(k_i, \alpha_i; \boldsymbol{q}, d\big)_{(\hat{x}_i, x)} \qquad \text{(C.22)}
$$

Where, the second equality follows by taking k_i, α_i and $q(\hat{x}_i | x)^{\alpha_i}$ out of the product over i, using the multivariate notation (4.3), and the last equality defines G. To summarize,

$$
\frac{\partial^{\alpha_0}}{\partial \beta^{\alpha_0}} q(\hat{x} | x)^{\boldsymbol{\alpha}_+} = \boldsymbol{\alpha}! \, q(\hat{x} | x)^{\boldsymbol{\alpha}_+} \sum_{\boldsymbol{k}: \, |\boldsymbol{k}| = \alpha_0} \prod_{i=1}^M G\big(k_i, \alpha_i; \boldsymbol{q}, d\big)_{(\hat{x}_i, x)} \qquad \text{(C.23)}
$$

where $G\big(k, a; \boldsymbol{q}, d\big)$ is a function on integers $0 \le k, a$, whose values are $M \times N$ matrices. We set $G(k, a; \boldsymbol{q}, d) = 0$ if $a = 0 < k$ due to the comment after (C.19), and otherwise

$$
G\big(k, a; \boldsymbol{q}, d\big)_{(\hat{x}, x)} := \sum_{\substack{\boldsymbol{t}: \, |\boldsymbol{t}| \le a, \\ \sum_j j \cdot t_j = k}} \frac{1}{\boldsymbol{t}! \, (a - |\boldsymbol{t}|)!} \prod_{j=1}^k \left(\frac{P_j(\hat{x}, x)}{j!} \right)^{t_j} \qquad \text{(C.24)}
$$

where $\boldsymbol{t} \in \mathbb{N}_0^k$. We write below $G(k, a)$ for short. It depends on $\boldsymbol{q}$ and the distortion d since so does $P_j(\hat{x}, x)$ (2.29). It might sometimes be more convenient computationally to have $\boldsymbol{t} \in \mathbb{N}_0^l$ for some pre-fixed value l. As $\boldsymbol{t}$ represents a partition of k, any integer $l \ge k$ would do.

To complete the calculation at (C.16), denote $\boldsymbol{\alpha}'_+ := \boldsymbol{\alpha}_+ + \boldsymbol{e}_{\hat{x}'}$ for $\boldsymbol{\alpha}$ with 1 added to its $\hat{x}'$-coordinate, where $\boldsymbol{e}_{\hat{x}'} \in \mathbb{R}^M$ is the standard basis vector at $\hat{x}'$. In coordinates, $\alpha'_i = \alpha_i + \delta_{i, \hat{x}'}$. This

does not affect the β coordinate $\alpha_0 = \alpha'_0$. Also, note that $q(\hat{x}'|x) \cdot q(\hat{x}|x)^{\boldsymbol{\alpha}+} = q(\hat{x}|x)^{\boldsymbol{\alpha}+ + e_{\hat{x}'}} = q(\hat{x}|x)^{\boldsymbol{\alpha}'_+}$, where $q(\hat{x}|x)^{\boldsymbol{\alpha}+}$ is considered as an $\hat{x}$-indexed vector for x fixed, as before. Thus,

$$
\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial\beta^{\alpha_0}\,\partial\boldsymbol{r}^{\boldsymbol{\alpha}+}}q(\hat{x}'|x) \overset{\text{(C.16)}}{=} \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!}{\boldsymbol{r}^{\boldsymbol{\alpha}+}} \cdot \frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}}\left\{ q(\hat{x}|x)^{\boldsymbol{\alpha}+}\left[\alpha_{\hat{x}'} - |\boldsymbol{\alpha}_+|\cdot q(\hat{x}'|x)\right]\right\}
$$

$$
\overset{\text{(C.23)}}{=} \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!}{\boldsymbol{r}^{\boldsymbol{\alpha}+}} \cdot \left\{ \alpha_{\hat{x}'}\cdot\boldsymbol{\alpha}!\, q(\hat{x}|x)^{\boldsymbol{\alpha}+} \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0}\prod_{i=1}^{M} G\bigl(k_i,\alpha_i\bigr)_{(\hat{x}_i,x)} \right.
$$

$$
\left. -|\boldsymbol{\alpha}_+|\cdot\boldsymbol{\alpha}'!\, q(\hat{x}|x)^{\boldsymbol{\alpha}'_+} \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0}\prod_{i=1}^{M} G\bigl(k_i,\alpha'_i\bigr)_{(\hat{x}_i,x)} \right\} \qquad \text{(C.25)}
$$

Where we applied formula (C.23) for $\frac{\partial^{\alpha_0}}{\partial\beta^{\alpha_0}}q(\hat{x}|x)^{\boldsymbol{\alpha}+}$ once to $q(\hat{x}|x)^{\boldsymbol{\alpha}+}$ and once to $q(\hat{x}'|x)\cdot q(\hat{x}|x)^{\boldsymbol{\alpha}+}$. From the definition (4.3), $\boldsymbol{\alpha}' ! = (1 + \alpha_{\hat{x}'})\cdot\boldsymbol{\alpha}!$. Proceeding with the calculation,

$$
= \frac{(-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!}{\boldsymbol{r}^{\boldsymbol{\alpha}+}} \cdot \boldsymbol{\alpha}!\, q(\hat{x}|x)^{\boldsymbol{\alpha}+} \cdot \left\{ \alpha_{\hat{x}'}\cdot \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0}\prod_{i=1}^{M} G\bigl(k_i,\alpha_i\bigr)_{(\hat{x}_i,x)} \right.
$$

$$
\left. -|\boldsymbol{\alpha}_+|\cdot(1+\alpha_{\hat{x}'})\cdot q(\hat{x}'|x) \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0}\prod_{i=1}^{M} G\bigl(k_i,\alpha_i + \delta_{i,\hat{x}'}\bigr)_{(\hat{x}_i,x)} \right\}
$$

$$
= (-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)!\,\boldsymbol{\alpha}! \cdot \left(\frac{q(\hat{x}|x)}{r(\hat{x})}\right)^{\boldsymbol{\alpha}+} \cdot \sum_{\boldsymbol{k}:\,|\boldsymbol{k}|=\alpha_0}\left(\prod_{i\neq\hat{x}'} G\bigl(k_i,\alpha_i\bigr)_{(\hat{x}_i,x)}\right)
$$

$$
\cdot\left\{ \alpha_{\hat{x}'}\cdot G\bigl(k_{\hat{x}'},\alpha_{\hat{x}'}\bigr)_{(\hat{x}',x)} - |\boldsymbol{\alpha}_+|\cdot(1+\alpha_{\hat{x}'})\cdot q(\hat{x}'|x)\cdot G\bigl(k_{\hat{x}'},1+\alpha_{\hat{x}'}\bigr)_{(\hat{x}',x)} \right\} \quad \text{(C.26)}
$$

Where, the last equality follows since the two products with G are identical at all but the $\hat{x}'$ multiplicand, so that $\prod_{i\neq\hat{x}'} G\bigl(k_i,\alpha_i\bigr)_{(\hat{x}_i,x)}$ can be taken out of the curly brackets. This is formula (5.15), completing the proof. $\qquad\square$

C.5 Proof of Proposition 23, Blahut-Arimoto's Jacobian in encoder coordinates

Unlike the previous subsections in Section C which take the input marginal $\boldsymbol{r}$ as the variable, here we consider the encoder $\boldsymbol{q}$ as the variable.

Proof of Proposition 23. We re-state the Blahut-Arimoto equations, with the encoder now playing the role of input and output distributions. Starting at the i-th iteration with $p_i(\hat{x}|x)$, set

$$
p_i(\hat{x}) := \sum_x p_X(x)p_i(\hat{x}|x) . \qquad \text{(C.27)}
$$

Then, output

$$
p_{i+1}(\hat{x}|x) := \frac{p_i(\hat{x})e^{-\beta d(x,\hat{x})}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}} . \qquad \text{(C.28)}
$$

For particular input coordinates $\hat{x}', x'$ and output coordinates $\hat{x}, x$, we would like to calculate

$$
\frac{\partial p_{i+1}(\hat{x}|x)}{\partial p_i(\hat{x}'|x')} = \sum_{\hat{x}''} \frac{\partial p_{i+1}(\hat{x}|x)}{\partial p_i(\hat{x}'')} \frac{\partial p_i(\hat{x}'')}{\partial p_i(\hat{x}'|x')} , \qquad \text{(C.29)}
$$

where the equality is due to the multivariate chain rule.

For the first integrand at (C.29),

$$\frac{\partial p_{i+1}(\hat{x}|x)}{\partial p_i(\hat{x}'')} \overset{(C.28)}{=} \frac{\partial}{\partial p_i(\hat{x}'')} \frac{p_i(\hat{x})e^{-\beta d(x,\hat{x})}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}} =$$

$$= \frac{\delta_{\hat{x},\hat{x}''}\ e^{-\beta d(x,\hat{x})}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}} - \frac{p_i(\hat{x})e^{-\beta d(x,\hat{x})}}{(\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')})^2} \cdot \overbrace{\frac{\partial}{\partial p_i(\hat{x}'')} \sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}}^{e^{-\beta d(x,\hat{x}'')}} =$$

$$= \frac{e^{-\beta d(x,\hat{x}'')}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}} \left[\delta_{\hat{x},\hat{x}''} - \frac{p_i(\hat{x})e^{-\beta d(x,\hat{x})}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}}\right] \overset{(C.28)}{=}$$

$$= \frac{e^{-\beta d(x,\hat{x}'')}}{\sum_{\hat{x}'} p_i(\hat{x}')e^{-\beta d(x,\hat{x}')}} \left[\delta_{\hat{x},\hat{x}''} - p_{i+1}(\hat{x}|x)\right] \quad \text{(C.30)}$$

For the second,

$$\frac{\partial p_i(\hat{x}'')}{\partial p_i(\hat{x}'|x')} \overset{(C.27)}{=} \frac{\partial}{\partial p_i(\hat{x}'|x')} \sum_x p_X(x)p_i(\hat{x}''|x) = \delta_{\hat{x}',\hat{x}''}\ p_X(x') \quad \text{(C.31)}$$

Combining the results,

$$\frac{\partial p_{i+1}(\hat{x}|x)}{\partial p_i(\hat{x}'|x')} \overset{(C.29)}{=} \sum_{\hat{x}''} \frac{\partial p_{i+1}(\hat{x}|x)}{\partial p_i(\hat{x}'')} \frac{\partial p_i(\hat{x}'')}{\partial p_i(\hat{x}'|x')} \overset{(C.30)}{\underset{(C.31)}{=}}$$

$$= \sum_{\hat{x}''} \frac{e^{-\beta d(x,\hat{x}'')}}{\sum_{\hat{x}'''} p_i(\hat{x}''')e^{-\beta d(x,\hat{x}''')}} \left[\delta_{\hat{x},\hat{x}''} - p_{i+1}(\hat{x}|x)\right] \cdot \delta_{\hat{x}',\hat{x}''}\ p_X(x') =$$

$$= \frac{e^{-\beta d(x,\hat{x}')}}{\sum_{\hat{x}''} p_i(\hat{x}'')e^{-\beta d(x,\hat{x}'')}} \left[\delta_{\hat{x},\hat{x}'} - p_{i+1}(\hat{x}|x)\right] p_X(x') \quad \text{(C.32)}$$

As $\frac{p_{i+1}(\hat{x}'|x)}{p_i(\hat{x}')} = \frac{e^{-\beta d(x,\hat{x}')}}{\sum_{\hat{x}''} p_i(\hat{x}'')e^{-\beta d(x,\hat{x}'')}}$ by (C.28) whenever $p_i(\hat{x}') \neq 0$ this yields the result. $\qquad \square$

D Proofs for error analysis

The following lemma is used for error analysis and for assessing computational costs of RD derivative tensors. While, the rest of this section contains results used only for error analysis.

Lemma 33 (Bounds on the complexity of P_k)**.** *Each polynomial P_k (2.23)-(2.24) is of degree k at most, and can be written as a sum of at most $2^k k!$ monomials in the coefficients 1 and -1. In particular, P_k has no more than $2^k k!$ monomials.*

Proof of Lemma 33. First, for the degree, note that the first addend $(x_1 - x_0) \cdot P_k$ in the inductive definition (2.24) of P_{k+1} increases the degree by 1, while by the definition (2.22) of $\bar{d}$, deriving P_k increases its degree by 1 at most, $\deg \bar{d}P_k \leq 1 + \deg P_k$.

Second, the derivative of a monomial $x_{i_1} \cdot x_{i_2} \cdot \cdots \cdot x_{i_k}$ of degree k is a sum of at most k differentiations, each of which is a sum of two monomials. Explicitly, if neither variable x_{i_j} is x_0, then

$$\bar{d}\Big(\prod_{j=1}^k x_{i_j}\Big) = \sum_{l=1}^k \Big(\prod_{j \neq l}^k x_{i_j}\Big)\bar{d}x_{i_l} \overset{(2.22)}{=} \sum_{l=1}^k \Big(\prod_{j \neq l}^k x_{i_j}\Big)(x_1 \cdot x_{i_l} - x_{i_l+1}) \quad \text{(D.1)}$$

which is a sum of at most $2k$ monomials. The sum is shorter if any i_j is 0, or if the monomial we've started with is of degree a smaller than k. Denote by l_k the minimal number of monomials in P_k, when represented as a sum with coefficients ± 1. By the inductive definition (2.24) of P_k,

$$l_{k+1} \leq 2 \cdot l_k + 2k \cdot l_k = 2(k+1) \cdot l_k . \tag{D.2}$$

Where, the first term $2 \cdot l_k$ bounds the number of monomials in $(x_1 - x_0) \cdot P_k$, and by (D.1), $2k \cdot l_k$ bounds that in $\bar{d}P_k$. Applying (D.2) inductively starting at $l_0 = 1$ (2.23), we obtain $l_k \leq (2 \cdot 1) \cdot (2 \cdot 2) \cdots (2 \cdot k) = 2^k k!$.

When the coefficients are not restricted to ± 1, then identical monomials can be grouped together, showing that the minimal number of monomials in P_k is smaller. $\square$

D.1 Proof that RD derivative tensors are bounded uniformly, Lemma 26

Proof of Lemma 26. We show that each of the quantities in Section 2.3 is bounded uniformly on the closed δ-interior of the simplex $\Delta[\hat{\mathcal{X}}]$, by a bound which depends only on the orders of differentiation b and m, and on the problem's properties, via d and p_X.

To synchronize the Lemma's notation at (7.9) with the explicit forms (2.30) and (2.31) of the derivative tensors of $Id - BA_\beta$ (1.2) (Theorem 4), define $\boldsymbol{\alpha} := (\alpha_0, \boldsymbol{\alpha}_+) \in \mathbb{N}_0^{M+1}$ as following. Set $\alpha_0 := b$, the number of differentiations with respect to β. Next, using the tensor indices $(i_1, i_2, \ldots, i_m)$ at (7.9), define $\boldsymbol{\alpha}_+ \in \mathbb{N}_0^M$ by $\boldsymbol{\alpha}_+ := \boldsymbol{e}_{i_1} + \boldsymbol{e}_{i_2} + \cdots + \boldsymbol{e}_{i_m}$, where $\boldsymbol{e}_j$ is the standard j-th basis vector. Carefully note that $|\boldsymbol{\alpha}_+| \overset{(4.3)}{=} m$ by definition, and so $|\boldsymbol{\alpha}| = m + b$. cf., the comments after Equation (2.6), for the two different notations of high-order derivatives.

Next, from definition (2.28),

$$\mathbb{E}_{q(\hat{x}'|x)} \left[d(x, \hat{x}')^k \right] = \sum_{\hat{x}'} q(\hat{x}'|x) d(x, \hat{x}')^k \leq \sum_{\hat{x}'} q(\hat{x}'|x) d_{max}^k = d_{max}^k \tag{D.3}$$

for *any* conditional distribution $q(\hat{x}|x)$. Therefore, by the definition (2.29) of the matrices $P_k[\boldsymbol{q}; d]$, each of its entries $(\hat{x}, x)$ is bounded,

$$|P_k[\boldsymbol{q}; d](\hat{x}, x)| \leq 2^k k! \cdot (d_{max}^k)^k . \tag{D.4}$$

For, by Lemma 33, each P_k can be written as a sum of no more than $2^k k!$ monomials, each of degree k at most, with the value of each variable bounded by (D.3). This immediately shows that repeated partial β-derivatives are uniformly bounded on the entire simplex,

$$\left| \frac{\partial^{\alpha_0}}{\partial \beta^{\alpha_0}} \left(Id - BA_\beta \right)[\boldsymbol{r}](\hat{x}) \right| \overset{(2.30)}{\leq} \sum_x p_X(x) q(\hat{x}|x) \cdot |P_{\alpha_0}(\hat{x}, x)| \overset{(D.4)}{\leq} 2^{\alpha_0} \alpha_0! \cdot d_{max}^{\alpha_0^2} \tag{D.5}$$

For mixed derivatives, first note that for each k, a,

$$\left| G(k, a)_{(\hat{x}, x)} \right| \overset{(2.32)}{\leq} \sum_{\substack{\boldsymbol{t}: \, |\boldsymbol{t}| \leq a, \\ \sum_j j \cdot t_j = k}} 1 \cdot \prod_{j=1}^{k} \left(\frac{|P_j(\hat{x}, x)|}{j!} \right)^{t_j} \overset{(D.4)}{\leq} p(k) \cdot \left(2^k k! \cdot d_{max}^{k^2} \right)^a \tag{D.6}$$

Where, $p(k)$ stands for the partition function (the number of integer partitions of k), and we have discarded the factorials at each denominator. For the last inequality, note that $\sum_j t_j \leq a$.

Next, by our assumption that $r(\hat{x}') \geq \delta$ for all $\hat{x}'$ and the definition (4.3) of vector power, it follows that $(1/r(\hat{x}'))^{\boldsymbol{\alpha}_+} \leq \frac{1}{\delta^{|\boldsymbol{\alpha}_+|}}$. Each coordinate $\alpha_{\hat{x}}$ of $\boldsymbol{\alpha}_+$ is bounded by $|\boldsymbol{\alpha}_+|$, and so $\boldsymbol{\alpha}!$ is

bounded by $\alpha_0 \cdot |\boldsymbol{\alpha}_+|!^M$. This allows to bound $\left| (-1)^{|\boldsymbol{\alpha}_+|-1}(|\boldsymbol{\alpha}_+|-1)! \, \boldsymbol{\alpha}! \sum_x p_X(x) \left(\frac{q(\hat{x}'|x)}{r(\hat{x}')} \right)^{\boldsymbol{\alpha}_+} \right|$
in (2.31) from above by $\frac{(|\boldsymbol{\alpha}_+|-1)! \, \alpha_0 \, |\boldsymbol{\alpha}_+|!^M}{\delta^{|\boldsymbol{\alpha}_+|}}$. Similarly, each coordinate $k_{\hat{x}}$ of $\boldsymbol{k} \in \mathbb{N}_0^M$ with $|\boldsymbol{k}| = \alpha_0$
is bounded by α_0, and so $|G(k, 1+a)|$ can be bounded by evaluating the upper bound (D.6) at
$(\alpha_0, 1 + |\boldsymbol{\alpha}_+|)$. From the combinatorial definition of the binomial coefficient, there are $\binom{\alpha_0 + M - 1}{\alpha_0}$
integral vectors $\boldsymbol{k} \in \mathbb{N}_0^M$ with $|\boldsymbol{k}| = \alpha_0$. Using these, the formula (2.31) for the mixed derivatives
can be bounded by,

$$\left| \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial \beta^{\alpha_0} \partial \boldsymbol{r}^{\boldsymbol{\alpha}_+}} (Id - BA_\beta)[\boldsymbol{r}](\hat{x}) \right|$$

$$\stackrel{(2.31)}{\leq} 1 + \frac{2(|\boldsymbol{\alpha}_+|+1)! \, \alpha_0}{\delta^{|\boldsymbol{\alpha}_+|}} \binom{\alpha_0 + M - 1}{\alpha_0} \left[|\boldsymbol{\alpha}_+|! \, p(\alpha_0) \cdot \left(2^{\alpha_0} \alpha_0! \cdot d_{max}^{\alpha_0^2} \right)^{1+|\boldsymbol{\alpha}_+|} \right]^M \quad \text{(D.7)}$$

Since (D.7) bounds the right-hand side of (D.5) from above, this completes the proof. $\qquad \square$

D.2   Proof that Taylor method converges between RD bifurcations, Theorem 5

Proof of Theorem 5. In this proof, denote $\partial_\delta \Delta[\hat{\mathcal{X}}] := \{ p \in \Delta[\hat{\mathcal{X}}] : \exists \hat{x} \; p(\hat{x}) \leq \delta \}$ for the closed
δ-boundary of the simplex, and $\overline{\Delta[\hat{\mathcal{X}}] \setminus \partial_\delta \Delta[\hat{\mathcal{X}}]}$ for the closed δ-interior.

For the first claim of this Theorem, set $\beta_f(\delta)$ to be the β value of the first time that $\boldsymbol{r}_\beta$ reaches
the δ-boundary. That is, the largest β such that $\beta < \beta_0$ and $\boldsymbol{r}_\beta \in \partial_\delta \Delta[\hat{\mathcal{X}}]$. This is well defined,
since $\partial_\delta \Delta[\hat{\mathcal{X}}]$ is compact and $\boldsymbol{r}_\beta$ is a continuous function of β, by Assumption 2 in Section I.2.1.
If $\boldsymbol{r}_\beta$ never reaches the δ-boundary of $\Delta[\hat{\mathcal{X}}]$ for $\beta \in [0, \beta_0]$ then set $\beta_f(\delta) = 0$.

For the second claim, we set to prove that the conditions of Theorem 25 in Section II.7.1 are
met; namely, of error analysis for an l-th order Taylor method. To invoke that Theorem, it suffices
to show (i) that the derivative's norm $\left\| \frac{d^{l+1} \boldsymbol{r}_\beta}{d\beta^{l+1}} \right\|_\infty$ at the true solution $\boldsymbol{r}_\beta$ is bounded uniformly on
$[\beta_f(\delta), \beta_0]$, and (ii) that the Taylor polynomial T_l (7.4) has a finite Lipschitz constant L_l. The
assumptions of that Theorem require that the Lipschitz condition holds for any $\beta \in [\beta_f(\delta), \beta_0]$ and
$\tilde{\boldsymbol{r}} \in \mathbb{R}^M$. However, the condition $\tilde{\boldsymbol{r}} \in \mathbb{R}^M$ may be relaxed to only requiring that $\tilde{\boldsymbol{r}}$ is not too far
away from a true solution $\boldsymbol{r}_\beta$, so long that the step-size will eventually be taken to be small enough,
[Butcher, 2016, Working assumption 210A]. We may choose δ' with $0 < \delta' < \delta$, and prove that the
Lipschitz condition (ii) holds for $(\tilde{\boldsymbol{r}}, \beta)$ with $\beta \in [\beta_f(\delta), \beta_0]$ and $\|\tilde{\boldsymbol{r}} - \boldsymbol{r}_\beta\|_\infty \leq \eta := \delta - \delta'$. Namely,
we may consider only points in a "tube" $T_\eta := \{ (\tilde{\boldsymbol{r}}, \beta) : \beta \in [\beta_f(\delta), \beta_0] \text{ and } \|\tilde{\boldsymbol{r}} - \boldsymbol{r}_\beta\|_\infty \leq \eta \}$ around
the true solution $\boldsymbol{r}_\beta$. For $\eta = 0$, T_0 is simply the graph of $\boldsymbol{r}_\beta$. Note that by the first claim, if
$0 < \eta < \delta$, then the $\tilde{\boldsymbol{r}}$ coordinate of T_η is contained in the closed δ'-interior, and thus in the interior
of the simplex. T_η is then essentially the product of two compact spaces, the closed η-ball (around
$\boldsymbol{r}_\beta \in \Delta[\hat{\mathcal{X}}]$) and the interval $[\beta_f(\delta), \beta_0]$.

Write $J(\tilde{\boldsymbol{r}}, \beta) := D_{\boldsymbol{r}}(Id - BA_\beta)|_{(\tilde{\boldsymbol{r}}, \beta)}$ for the Jacobian matrix. Its general form (*not* necessarily
at a fixed point) at a distribution $\tilde{\boldsymbol{r}}$ of full support is given by Corollary 10 in Section II.5.2 (formula
(5.7) there). If $\boldsymbol{r}_\beta$ is in addition a fixed point of BA_β, then the basic properties of $J(\boldsymbol{r}_\beta, \beta)$ are
given by Theorem 11 there. In particular, it is non-singular so long that $\boldsymbol{r}_\beta$ is in the simplex
interior, which by the first claim holds for $\beta \in [\beta_f(\delta), \beta_0]$.

Before proving (i) and (ii), we shall show that $\eta > 0$ can be chosen small enough, such that
$J(\tilde{\boldsymbol{r}}, \beta)$ is non-singular for every $(\tilde{\boldsymbol{r}}, \beta) \in T_\eta$. From this, it shall follow that the matrix norm
$\|J(\tilde{\boldsymbol{r}}, \beta)^{-1}\|_\infty$ of its inverse is well-defined on T_η, and thus bounded uniformly. For, it is a continu-
ous real-valued function on the compact set T_η, and so obtains a maximal value. e.g., [Ortega, 1990,
1.3] for matrix norms. When $\tilde{\boldsymbol{r}}$ is set to the true solution $\boldsymbol{r}_\beta$, this is rather straightforward from

Theorem 11. While for $\tilde{r}$ slightly off a true solution, this follows from continuity and compactness, as we show next. e.g., [Munkres, 2000, Chapter 3] on compactness.

First, we prove that J is non-singular when evaluated at the true solution $\boldsymbol{r}_\beta$. This follows since the composition

$$\beta \mapsto \boldsymbol{r}_\beta \mapsto J(\boldsymbol{r}_\beta, \beta) \mapsto |\det J(\boldsymbol{r}_\beta, \beta)| \tag{D.8}$$

is continuous on $[\beta_f(\delta), \beta_0]$ (shown below), and so obtains a minimal value $d' \geq 0$ at some point β' there. $J(\boldsymbol{r}_{\beta'}, \beta')$ is non-singular by the Jacobian's properties mentioned above, and so d' is strictly positive. In particular, d' does not depend on the value of $\eta \geq 0$. This shows that $J(\boldsymbol{r}_\beta, \beta)$ is non-singular on $[\beta_f(\delta), \beta_0]$. Each function in the composition (D.8) is indeed continuous. For the first $\beta \mapsto \boldsymbol{r}_\beta$ this is by Assumption 2. For the second $\boldsymbol{r}_\beta \mapsto J(\boldsymbol{r}_\beta, \beta)$, by formula (5.7) of Corollary 10, the entries of $J(\tilde{r}, \beta)$ are continuous in both β and $\tilde{r}$, so long that $\tilde{r}$ is in the interior of $\Delta[\hat{\mathcal{X}}]$, which holds at $\boldsymbol{r}_\beta$ by the first claim. Finally, the determinant of a matrix is continuous, as it is a sum of products of matrix entries.

Second, we show that $\eta > 0$ can be chosen small enough that J is non-singular also at points in T_η other than the true solution. Fix some η with $0 < \eta < \delta$, and define a function f on T_η by

$$f(\tilde{r}, \beta) := |\det J(\tilde{r}, \beta)| \tag{D.9}$$

By the note after T_η's definition, its projection onto $\Delta[\hat{\mathcal{X}}]$ is in the interior of the simplex. Hence, f is well-defined and continuous in $(\tilde{r}, \beta)$, by formula (5.7). It satisfies $f(\boldsymbol{r}_\beta, \beta) \geq d' > 0$ by the argument above. By the definition of continuity, [Munkres, 2000, §18], the inverse image $f^{-1}((d'/2, \infty])$ is open in T_η; it contains T_0 (the graph of $\boldsymbol{r}_\beta$). As T_η is a product of compact spaces, then by the tube lemma [Munkres, 2000, Lemma 26.8] there is $0 < \eta' \leq \eta$ such that $T_0 \subset T_{\eta'} \subseteq f^{-1}((d'/2, \infty])$. To see this, note that any open neighborhood of 0 in the η-ball contains an open η'-ball around 0, from the definition of a basis for a topology. Therefore, $|\det J|$ is at least $d'/2 > 0$ on $T_{\eta'}$, as argued.

Summarizing the above, we have shown that $0 < \eta' < \delta$ can be chosen such that $J(\tilde{r}, \beta)$ is non-singular on $T_{\eta'}$. Therefore, matrix inversion is well-defined on it. As matrix inversion and norm are continuous, then $\|J(\tilde{r}, \beta)^{-1}\|_\infty$ is continuous on the compact set $T_{\eta'}$, and so obtains a maximum value on it. That is, the Jacobian's inverse is of uniformly bounded matrix norm, for distributions $\tilde{r}$ at most η'-far from $\boldsymbol{r}_\beta$.

Aided by the above, we turn to prove (i) and (ii). For (i), let $\frac{d^l \tilde{r}}{d\beta^l}$ denote the numerical derivative calculated from formula (2.17) (Theorem 3 in Section I.2.2) at an approximation $\tilde{r}$ of the true solution $\boldsymbol{r}_\beta$, such that $\tilde{r}$ is at most η'-far from $\boldsymbol{r}_\beta$. We prove by induction on $l > 0$ that its norm $\left\|\frac{d^l \tilde{r}}{d\beta^l}\right\|_\infty$ is bounded uniformly on $T_{\eta'}$. Since $T_{\eta'}$ contains the graph of $\boldsymbol{r}_\beta$, this shall suffice to prove (i). Assume that it holds for any $0 < k < l$, for the norms $\left\|\frac{d^k \tilde{r}}{d\beta^k}\right\|_\infty$. We would like to prove that so is the l-th derivative $\left\|\frac{d^l \tilde{r}}{d\beta^l}\right\|_\infty$. The assumption is of course vacuous when $l = 1$. Indeed, the first-order implicit derivative $\frac{d\tilde{r}}{d\beta}$ is the only one that does not involve implicit derivatives of lower order in its calculation.

For the induction step, note that the l-th derivative $\frac{d^l \tilde{r}}{d\beta^l}$ involves derivative tensors $D^m_{\beta^b, \boldsymbol{r}^{m-b}}(Id - BA_\beta)|_{(\tilde{r}, \beta)}$, with $0 \leq b \leq m \leq l$; see Equation (2.17) (Theorem 3). By Lemma 26 in Section II.7.2, the entries of these tensors are bounded uniformly on $\overline{\Delta[\hat{\mathcal{X}}] \setminus \partial_{\delta'} \Delta[\hat{\mathcal{X}}]}$ (regardless of β), for $\delta' = \delta - \eta' > 0$, which in turn contains $\tilde{r} \in T_{\eta'}$. By the induction's hypothesis, the lower-order derivatives are also bounded uniformly on $T_{\eta'}$. Therefore, the right-hand side of (2.17) is bounded uniformly, as it involves only a (particular) finite sum of these tensors, which are evaluated at implicit derivatives of bounded coordinates. That is, it is a sum of products of quantities that are bounded uniformly on $T_{\eta'}$. To complete the induction step, both sides of (2.17) need to be

multiplied by the inverse Jacobian. Yet, $\|(D_{\boldsymbol{r}}(Id - BA_\beta)|_{(\tilde{\boldsymbol{r}},\beta)})^{-1}\|_\infty$ is bounded uniformly on $T_{\eta'}$ (as shown before), and thus so is $\left\|\frac{d^l\tilde{\boldsymbol{r}}}{d\beta^l}\right\|_\infty$, as required.

For (ii), Lipschitz continuity can be established by means of a supremum over the derivative's matrix norm, e.g., Equation (3.9) at Atkinson et al. [2011],

$$L_l := \sup \|D_{\boldsymbol{r}} T_l\|_\infty , \tag{D.10}$$

where the supremum is over $T_{\eta'}$. A differentiable functions T_l is Lipschitz continuous if the supremum (D.10) is finite.

From the definition (7.4) of Taylor method,

$$D_{\boldsymbol{r}} T_l(\tilde{\boldsymbol{r}}, \beta, \Delta\beta) = \frac{1}{1!} \cdot D_{\boldsymbol{r}} \frac{d\tilde{\boldsymbol{r}}}{d\beta} + \frac{\Delta\beta}{2!} \cdot D_{\boldsymbol{r}} \frac{d^2\tilde{\boldsymbol{r}}}{d\beta^2} + \cdots + \frac{\Delta\beta^{l-1}}{l!} \cdot D_{\boldsymbol{r}} \frac{d^l\tilde{\boldsymbol{r}}}{d\beta^l} \tag{D.11}$$

As matrix norms are sub-additive,

$$\|D_{\boldsymbol{r}} T_l(\tilde{\boldsymbol{r}}, \beta, \Delta\beta)\|_\infty \leq \frac{1}{1!} \cdot \left\|D_{\boldsymbol{r}} \frac{d\tilde{\boldsymbol{r}}}{d\beta}\right\|_\infty + \frac{|\Delta\beta|}{2!} \cdot \left\|D_{\boldsymbol{r}} \frac{d^2\tilde{\boldsymbol{r}}}{d\beta^2}\right\|_\infty + \cdots + \frac{|\Delta\beta|^{l-1}}{l!} \cdot \left\|D_{\boldsymbol{r}} \frac{d^l\tilde{\boldsymbol{r}}}{d\beta^l}\right\|_\infty \tag{D.12}$$

Thus, to prove that T_l has a finite Lipschitz constant, it suffices to show that the matrix norms of the matrices $D_{\boldsymbol{r}} \frac{d^k\tilde{\boldsymbol{r}}}{d\beta^k}$ are bounded uniformly on $T_{\eta'}$. By Proposition 27 in Section II.7.3, the Jacobian $D_{\boldsymbol{r}} \frac{d^l\tilde{\boldsymbol{r}}}{d\beta^l}$ for $l > 0$ can be expressed using derivative tensors $D^m_{\beta^b, \boldsymbol{r}^{m-b}}(Id - BA_\beta)|_{(\tilde{\boldsymbol{r}},\beta)}$ with $0 \leq b \leq m \leq l$, derivatives $\frac{d^k\tilde{\boldsymbol{r}}}{d\beta^k}$ of lesser or equal degree $0 < k \leq l$, Jacobians $D_{\boldsymbol{r}} \frac{d^k\tilde{\boldsymbol{r}}}{d\beta^k}$ of strictly lower degree $0 < k < l$, and the inverse-Jacobian $\left(D_{\boldsymbol{r}}(Id - BA_\beta)|_{(\tilde{\boldsymbol{r}},\beta)}\right)^{-1}$. As in the proof of (i), all these quantities are bounded uniformly on $T_{\eta'}$, showing that (D.12) is indeed bounded uniformly, as required. □

E   Proof for the complexity of root-tracking and of RD root-tracking

We analyze the computational and memory complexities of root tracking in general and of root tracking for RD problems in particular. As the complexity of Taylor's method is determined by its complexity at a point, this boils down to analyzing the complexity of the Algorithm 1 — both of its general form and of its specialization to RD.

This section is structured as follows. In Section E.1, we analyze the cost of the recursive formula (2.17) for implicit derivatives (Algorithm 1), assuming that the computational costs of the derivative tensors $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ are given. When derivative tensors are computationally expensive (as in RD), it might be preferable to memorize them, so as to avoid computing a tensor more than once. The complexity is discussed both with and without tensor memorization, at Propositions 28, 29 and 30. In Section E.2, we analyze the complexity of the various quantities needed to compute RD derivative tensors, those of $Id - BA_\beta$ (1.2). The results are combined in Section E.3, proving the upper bounds for the complexity of root tracking for RD, Theorem 6.

The upper bounds we provide are often loose. In practice, some of the memory and computational costs grow at a much lower rate. cf., Figures E.1 and E.2 for example. For actual computational measures timed with our implementation see Figure 1.2.

E.1 Complexity of root tracking: implicit derivatives of operator roots

We consider the complexity of the recursive formula (2.17) for $\frac{d^l \boldsymbol{x}}{d\beta^l}$ (Theorem 3). Note that its right-hand side is comprised of an outer and of an inner summation. The outer summation is over the $p(l) - 1$ non-trivial partitions, while the inner summation is over the number of parts of size 1 in a given partition, $M_1 \cdot \delta(p_1 = 1)$. That is, the inner one is over the multiplicity of 1 in a partition. For example, 1 is of multiplicity 3 in the partition $1 + 1 + 1$ of 3, of multiplicity 1 in $2 + 1$, and of multiplicity 0 in the trivial partition 3.

Integer partitions can be grouped by their multiplicity of 1. For, "set aside" a single part of size 1, and consider partitions of $l - 1$:

$$l = 1 + \underbrace{(\ldots \quad \ldots)}_{\text{a partition of } l-1} \tag{E.1}$$

Any partition of l with at least one part of size 1 can be written as "$1+$ a partition of $(l-1)$", as in (E.1). Thus, there are exactly $p(l-1)$ partitions of l in which 1 is of multiplicity ≥ 1. Hence, there are $p(l) - p(l-1)$ partitions with no part of size 1. Setting aside two parts of size 1 shows that there are exactly $p(l-2)$ partitions of l in which 1 is of multiplicity ≥ 2. Hence, there are $p(l-1) - p(l-2)$ partitions which have exactly one part of size 1. Proceeding in this manner, we have

Lemma 34. *For $0 \leq j \leq l$, the number of partitions of l with exactly j parts of size 1 is*

$$p(l-j) - p(l-j-1) , \tag{E.2}$$

where $p(-1)$ is defined to be 0, and $p(0) := 1$.

A partition with j parts of size 1 contributes $j+1$ summands to the inner summation at (2.17). Grouping partitions by the multiplicity j of 1 in them, the total number of summands at the right-hand side of (2.17) is

$$1 \cdot \Big(p(l) - p(l-1)\Big) + 2 \cdot \Big(p(l-1) - p(l-2)\Big) + \cdots + l \cdot \Big(p(1) - p(0)\Big) + (l+1) \cdot p(0) - 1$$
$$= p(l) + p(l-1) + \cdots + p(1) + p(0) - 1 . \tag{E.3}$$

Lemma 35. *The number of summands at the right-hand side of (2.17) is $\sum_{j=0}^{l} p(j) - 1$.*

At least for our case of interest $F := Id - BA_\beta$ (1.2), the complexity of a derivative tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ is determined by m and b; see Section E.2. Thus, we would like to group partitions not only by the multiplicity of 1 in them, but also by their total multiplicity m. Denote $p_k(n)$ for the number of partitions of n to exactly k parts. From its definition, $p(n) = \sum_{k=0}^{n} p_k(n)$. For small k, n values, it can be calculated using the recurrence relation [Stanley, 2011, Section 1.7]

$$p_k(n) = p_{k-1}(n-1) + p_k(n-k) \tag{E.4}$$

and $p_0(0) = 1$; see Table 3 for example.

By the same reasoning as at (E.1), there are $p_{k-1}(l-1)$ partitions of l to k parts in which 1 is of multiplicity ≥ 1, $p_{k-2}(l-2)$ partitions of l to k parts in which 1 is of multiplicity ≥ 2, and so forth. Therefore, we have as before:

Lemma 36. *For $0 \leq j \leq k \leq l$, the number of partitions of l to k parts of which exactly j are of size 1 is*

$$p_{k-j}(l-j) - p_{k-j-1}(l-j-1) \stackrel{\text{(E.4)}}{=} p_{k-j}(l-k) , \tag{E.5}$$

where $p_k(n) := 0$ if either $k < 0$ or $n < 0$.

k, n	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1		1	1	1	1	1	1
2			1	1	2	2	3
3				1	1	2	3
4					1	1	2
5						1	1
6							1

Table 3: $p_k(n)$ for several small n, k values.

Write $C(b, m)$ for the complexity of calculating a derivative tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$, and $p_{\leq k}(n)$ for the number of partitions of n to at most k parts. By definition, $p_{\leq k}(n) = \sum_{i=0}^{k} p_i(n)$. Using Lemma 36, we rearrange the outer summation over partitions at (2.17). First, by the partitions' total multiplicity m (number of parts), and then by the number m_1 of parts of size 1. With this, the complexity of calculating the derivative tensors is

$$\sum_{m=0}^{l} \sum_{m_1=0}^{m} p_{m-m_1}(l-m) \sum_{b=0}^{m_1} C(b, m) = \sum_{m=0}^{l} \sum_{m_1=0}^{m} \sum_{b=0}^{m_1} p_{m-m_1}(l-m) C(b, m)$$

$$= \sum_{m=0}^{l} \sum_{b=0}^{m} \sum_{m_1=b}^{m} p_{m-m_1}(l-m) C(b, m) = \sum_{m=0}^{l} \sum_{b=0}^{m} C(b, m) \sum_{m_1=b}^{m} p_{m-m_1}(l-m)$$

$$= \sum_{m=0}^{l} \sum_{b=0}^{m} C(b, m) \sum_{j=0}^{m-b} p_j(l-m) = \sum_{m=0}^{l} \sum_{b=0}^{m} p_{\leq m-b}(l-m) C(b, m) \quad \text{(E.6)}$$

The second equality above follows since, for a given m, we are summing over all the integers b and m_1 with $0 \leq b \leq m_1 \leq m$. At the fourth equality we exchange m_1 with $j := m - m_1$. The last equality follows from the definition of $p_{\leq k}(n)$.

Since (E.6) adds up the complexity $C(b, m)$ once for each summand at the right-hand side of (2.17), then setting $C(b, m) := 1$ gives an alternative formula for the number of summands there. Combined with Lemma 35, we have

Corollary 37.

$$\sum_{m=0}^{l} \sum_{b=0}^{m} p_{\leq m-b}(l-m) = \sum_{j=1}^{l} p(j) . \quad \text{(E.7)}$$

Write $F = (F_1, F_2, \ldots, F_T)$ for the operator's coordinates, as at (2.6). On top of the complexity (E.6) for calculating the derivative tensors, we need to account for the complexity of evaluating the multilinear forms (2.6) they define, of summing the evaluated forms, and of finding a linear pre-image under $D_{\boldsymbol{x}} F$. Indeed, a tensor $D^m_{\beta^b, \boldsymbol{x}^{m-b}} F$ has T^{m-b+1} entries, and so its evaluation is $O((m-b+1) T^{m-b+1})$ operations; this is multiplied by the number of tensors of each rank, as in (E.6). An evaluated multilinear form has T entries. Thus, by Lemma 35, the complexity of their summation at the right of (2.17) is $O(T \cdot \sum_{j=1}^{l} p(j))$. Finding a linear pre-image is no more than $O(T^3)$ operations, e.g., using Gaussian elimination. This is summarized by Proposition 28 in Section II.8, on the complexity of l-th order derivative.

The computational cost is reduced drastically (at the expense of memory complexity) if every derivative tensor is computed only once and then memorized. By taking the all-ones partition

$1+1+\cdots+1$ of l, the l-th order formula (2.17) can be seen to contain *all* the l-th order derivative tensors: $D^l_{\beta^0,\boldsymbol{x}^l}F, D^l_{\beta^1,\boldsymbol{x}^{l-1}}F,\ldots,D^l_{\beta^l,\boldsymbol{x}^0}F$. These appear in the l-th order formula for the first time. For, the total multiplicity m cannot exceed the partitioned integer l, and so neither of the l-th order tensors can appear when formula (2.17) is used with orders $< l$. As $D^m_{\beta^b,\boldsymbol{x}^{m-b}}F$ has T^{m-b+1} coordinates, memorizing all the tensors used up to the l-th order of the recursive formula (2.17) boils down to memorizing

$$\sum_{m=0}^{l-1}\sum_{t=0}^{m}T^{t+1}\le l\cdot T^l \tag{E.8}$$

coordinates. Where, we've written t for $m-b$. We thus obtain Proposition 29 in Section II.8, on the complexity of the l-th order derivative with tensor memorization.

The cumulative computational and memory costs with memorization are summarized by Proposition 30 in Section II.8. Its proof is provided below.

Proof of Proposition 30. We sum the computational complexity (8.2) per l value (Proposition 29) over $l = 1,\ldots,L$.

For the first summand at (8.2), $\sum_{l=1}^{L}O(T^3) = O(L\cdot T^3)$. For the second,

$$\sum_{l=1}^{L}T\cdot\sum_{j=1}^{l}p(j) = T\cdot\sum_{l=1}^{L}\sum_{j=1}^{l}p(j) \le L^2T\cdot p(L) \le L^{5/4}Te^{\pi\sqrt{\frac{2}{3}L}} \tag{E.9}$$

Where, the first inequality follows by replacing $p(j)$ with the maximal summand $p(L)$, and the last follows from the upper bound $p(n) < \frac{e^{\pi\sqrt{2n/3}}}{n^{3/4}}$ at de Azevedo Pribitkin [2009].

Next, note that $p_{\le k}(n)$ is non-decreasing in n. For, to a partition $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_j$ of n to with $j \le k$ parts, one can injectively match the partition of $n+1$ with the same number of parts, defined by replacing λ_j with $\lambda_j + 1$. Thus,

$$p_{\le m-b}(l-m) \le p_{\le m-b}(L) \le p(L) \tag{E.10}$$

where the last inequality follow from the definition of $p_{\le k}(n)$. So, for the third summand at (8.2),

$$\sum_{l=1}^{L}\sum_{m=0}^{l}\sum_{b=0}^{m}p_{\le m-b}(l-m)O((m-b+1)T^{m-b+1}) = L\,p(L)\cdot\sum_{m=0}^{L}\sum_{b=0}^{m}O((m-b+1)T^{m-b+1})$$

$$= L\,p(L)\cdot\sum_{0\le t\le L}(L-t+1)O((t+1)T^{t+1}) = O(L^3T^{L+1}p(L)) = O(L^{9/4}T^{L+1}e^{\pi\sqrt{\frac{2}{3}L}}) \tag{E.11}$$

Where (E.10) was used at the first equality, carrying out the summation over l. Next, $m-b$ was replaced by t. The last equality again uses the bound at de Azevedo Pribitkin [2009].

For $L = 1$, the cost of the terms above adds up to

$$O(L\cdot T^3) + O(L^{5/4}Te^{\pi\sqrt{\frac{2}{3}L}}) + O(L^{9/4}T^{L+1}e^{\pi\sqrt{\frac{2}{3}L}}) = O(T^3)\,, \tag{E.12}$$

proving (8.5). While for $L \ge 2$,

$$O(L\cdot T^3) + O(L^{5/4}Te^{\pi\sqrt{\frac{2}{3}L}}) + O(L^{9/4}T^{L+1}e^{\pi\sqrt{\frac{2}{3}L}})$$

$$= O(L^{9/4}T^{L+1}e^{\pi\sqrt{\frac{2}{3}L}}) = O\left(e^{9/4\cdot\ln L+(L+1)\ln T+\pi\sqrt{2L/3}}\right) \tag{E.13}$$

which proves (8.4). □

E.2   Complexity of high-order derivative tensors of $Id - BA_\beta$

We analyze the complexity of the formulae for the derivative tensors $D_{\beta^b,\boldsymbol{x}^m}^{b+m}\left(Id - BA_\beta\right)[\boldsymbol{r}]\left(\hat{x}\right)$ (Theorem 4 in Section 2.3.). To that end, we distinguish between three kinds of computations: initial computations E.2.1 which depend neither on the RD problem at hand nor on the point of evaluation; initial computations E.2.2 at a particular point; and the cost of a derivative tensor E.2.3 given the above initial computations. The analysis roughly follows the order of the results' presentation in Section I.2.3, and the comments in II.5.5 on efficient computation of RD derivative tensors. The results below are summarized in Table 2, in Section II.8.

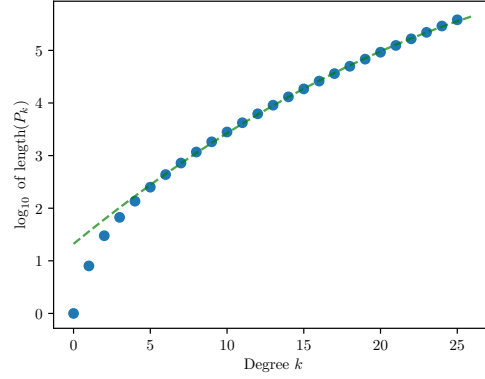### E.2.1   Problem-independent initial-computations

Figure E.1: Length of the default string representation of P_k used by our implementation. An approximation $-0.002739k^2 + 0.2377k + 1.323$ of the log-length is in dashed green.

The algebraic form of the polynomials P_k (2.23)-(2.24) needs to be computed only once and for all, as it depends neither on the problem details nor on the point of evaluation. We therefore ignore its computational complexity, and consider only its memory complexity.

By Lemma 33 (in Section D), each P_k is of degree k at most, and can be written as a sum of at most $2^k k!$ monomials, even when only 1 and -1 are allowed as coefficients. Each of these monomials can be encoded using $1 + (k+1)\log_2(k+1)$ bits: each of the $k+1$ variables is of degree $0, \ldots, k$, and one bit for the coefficient. Hence, the memory needed to store P_k is $O\left((k+1)\log_2(k+1)2^k k!\right)$.

In practice, the length of P_k is roughly 1.73^k, as demonstrated by Figure E.1. Our implementation stores the first 20 polynomials in about 110 kilobytes in compressed form (LZMA) and the first 25 polynomials in 544 KBs.

E.2.2   Initial computations at a point

Recall, we write $N := |\mathcal{X}|$ and $M := |\hat{\mathcal{X}}|$ for the source and reproduction alphabet sizes, respectively. Given an encoder $q(\hat{x}|x)$ of an RD problem defined by $\left(d(x,\hat{x}), p(x)\right)$, we evaluate the cost of the components needed to compute the M-by-N matrices $G(k, a; \boldsymbol{q})$ (2.32). These are indexed by $0 \le k \le L$ and $0 \le a \le 1 + L$, where L is the maximal order of derivation. Clearly, its memory complexity is $O(MN(1 + L)(2 + L))$.

The $P_k[\boldsymbol{q}; d]$ matrices (2.29) are needed to compute G; these require the expectations $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x,\hat{x}')^k\right]$ (2.28). The powers $d(x, \hat{x})^k$ may be computed only once per RD problem, and so we neglect their computational cost. Thus, the computation of $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x,\hat{x}')^k\right]$ costs $O(MN)$ operations, as it involves only multiplication by $q(\hat{x}|x)$ and a summation over $\hat{x}$. The zeroth variable $d(x,\hat{x})$ of P_k has MN coordinates, while the others $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x,\hat{x}')^k\right]$ have N coordinates, for $k \ge 1$. As P_k
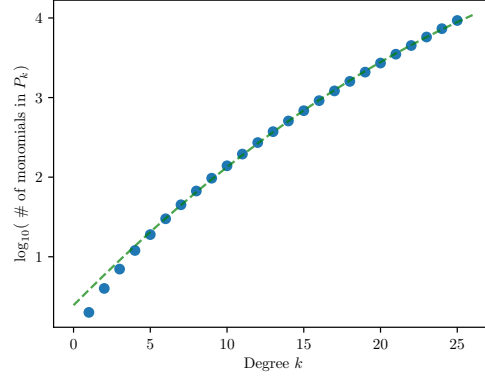
Figure E.2: The number of monomials in P_k. An approximation $-0.002084k^2 + 0.1944k + 0.3911$ of $\log_{10}(\# \text{ monomials})$ is in dashed green.

is of degree k at most, Lemma 33 in Section D, evaluating a single monomial at $P_k[\boldsymbol{q}; d]$ (2.29) is $O(MNk)$ operations. By the Lemma, P_k has no more than $2^k k!$ monomials, so the total cost is $O(MNk\, 2^k k!)$. In practice, there are roughly 1.56^k monomials in P_k, as seen in Figure E.2 for $k \leq 25$, and so the actual cost is much smaller.

For $G(k, a; \boldsymbol{q})$, computing $\left(\frac{P_j(\hat{x}, x)}{j!} \right)^{t_j}$ at (2.32) for a given $\boldsymbol{t}$ and j is $O(MN)$ operations. Accounting for the pointwise product over $j = 1, \ldots, k$, the cost for each $\boldsymbol{t}$ is $O(MNk)$. The entries of G can be computed by iterating only once over the partitions $\boldsymbol{t}$ of all the integers k with $0 \leq k \leq L$. For, given $\boldsymbol{t}$, an M-by-N integrand $\prod_j \left(\frac{P_j(\hat{x}, x)}{j!} \right)^{t_j}$ can be added to $G(k, a)$ for all $a \geq |\boldsymbol{t}|$, at $O(MNL)$ operations. That is, the entire cost for a particular $\boldsymbol{t}$ is $O(MNk) + O(MNL) = O(MNL)$, and so $O(MNL \sum_{k=0}^{L} p(k))$ when iterating over all the integer partitions $\boldsymbol{t}$.

E.2.3 Complexity of a derivative tensor

We analyze the cost of the various derivative tensors of $Id - BA_\beta$ (1.2), assuming that the initial computations in E.2.1 and E.2.2 were already done.

The memory complexity of $D_{\beta^b, \boldsymbol{x}^m}^{b+m}(Id - BA_\beta)[\boldsymbol{r}](\hat{x})$ (2.31) is $O(M^{m+1})$. The computational complexity of a derivative tensor $D_{\beta^b}^{b}(Id - BA_\beta)[\boldsymbol{r}](\hat{x})$ (2.30) with respect to β alone is $O(MN)$. For a mixed derivatives tensor $D_{\beta^b, \boldsymbol{x}^m}^{b+m}(Id - BA_\beta)[\boldsymbol{r}](\hat{x})$ with $m > 0$, fix $\boldsymbol{\alpha} = (b, \boldsymbol{\alpha}_+) \in \mathbb{N}_0^{M+1}$ with $\boldsymbol{\alpha}_+ \neq 0$. For any $\hat{x}$ and $\boldsymbol{k} \in \mathbb{N}_0^M$ with $|\boldsymbol{k}| = b$, the complexity of the integrand under the summation $\sum$ at (2.31) is $O(MN)$ operations. It is an N-vector. Summing over the $\binom{b+M-1}{b}$ choices for $\boldsymbol{k}$, the sum has a complexity of $O(\binom{b+M-1}{b} \cdot MN)$ operations. The other operations at (2.31) have a comparatively negligible cost. The above needs to be performed for each $\hat{x}$ and $\boldsymbol{\alpha}_+$ with $|\boldsymbol{\alpha}_+| = m$, which is $M \cdot \binom{m+M-1}{m}$ times. Thus, the complexity of the mixed partial derivatives formula (2.31) is $O(\binom{m+M-1}{m}\binom{b+M-1}{b} \cdot M^2 N)$ operations.

To a multi-index $\boldsymbol{\alpha}_+ \in \mathbb{N}_0^M$ may correspond multiple coordinates of a derivative tensor. For, a multi-index $\boldsymbol{\alpha}_+$ counts the number of derivations with respect to each of the M coordinates of $\boldsymbol{r}$, while a tensor entry stands for a particular order of the derivatives. e.g., for a reproduction alphabet of size $M = 2$, $\boldsymbol{\alpha}_+ = (2, 1)$ represents taking two derivatives with respect to r_1, and one with respect to r_2. To it, correspond the tensor entries $\frac{\partial^3}{\partial r_2 \partial r_1 \partial r_1}$, $\frac{\partial^3}{\partial r_1 \partial r_2 \partial r_1}$ and $\frac{\partial^3}{\partial r_1 \partial r_1 \partial r_2}$ at (2.6), standing respectively for the $(2, 1, 1), (1, 2, 1)$ and $(1, 1, 2)$ entries of a derivative tensor with $m = 3$. The number of tensor entries corresponding to a multi-index $\boldsymbol{\alpha}_+$ is $m!$ at most, with $O(M)$ copy operations for each tensor entry. Doing so for each $\boldsymbol{\alpha}_+$ is $O(\binom{m+M-1}{m} m! M)$ operations. While copying out partial derivatives to the various tensor entries need not be the most efficient solution, it is simple and straightforward to implement.

All in all, bounds on the complexities of RD derivative tensors are summarized in Table 2 in Section II.8.

E.3 Complexity of root tracking for RD

In this subsection we prove Theorem 6 on the complexities of root tracking for RD. We do so by compiling the results of Sections E.1 on the complexity of root-tracking and of E.2 on the complexity of RD derivative tensors (Table 2 in Section II.8). The proof boils down to adding the various costs associated with the derivatives at a point, up to order L.

Proof of Theorem 6. In Section E.2, we divided the cost of RD derivative tensors to initial computations and the calculation of the derivative tensors themselves. Using Table 2 in Section II.8, initial computations at a point require the algebraic form of P_k, the expectations $\mathbb{E}_{q(\hat{x}'|x)}\left[d(x,\hat{x}')^k\right]$ and the matrices $P_k[\boldsymbol{q};d]$ for $k = 0, \ldots, L$; and the matrices $G(k,a)$. Summing over the respective table elements, their memory cost is:

$$\sum_{k=0}^{L}\left[O\left(2^k k!(k+1)\log_2(k+1)\right) + O(N) + O(MN)\right] + O(MN(L+1)(L+2))$$

$$= O\left(2^L L!L^2 \ln L\right) + O(MNL^2) < O\left(e^{(L+5/2)\ln L - c_1 L + \ln \ln L + c_2}\right) + O(MNL^2)$$

$$< O\left(L^{(L+5/2)} \ln L\right) + O(MNL^2) \quad \text{(E.14)}$$

Where, the first expression at the first line is bounded by its maximal value at $k = L$, the first inequality follows from $n! \le e^{(n+1/2)\ln n - n + c_2}$ Robbins [1955], with $c_2 := \frac{1}{12} + \frac{1}{2}\ln(2\pi) \approx 1$, and by setting $c_1 := 1 - \ln 2 \approx 0.31$.

For the initial computations,

$$\sum_{k=0}^{L}\left[O(MN) + O(MNk\,2^k k!) + O(MNL\,p(k))\right]$$

$$= \sum_{k=0}^{L}\left[O(MNL\,2^k k!) + O(MNL\,p(k))\right] = \sum_{k=0}^{L} O(MNL\,2^k k!)$$

$$= O(MN2^L L!L^2) = O\left(MN \cdot e^{(L+5/2)\ln L - c_1 L + c_2}\right) \quad \text{(E.15)}$$

Where, k is bounded by L at the first equality, $O(2^k k!) + O(p(k)) = O(2^k k!)$ by the bound $p(k) < \frac{e^{\pi\sqrt{2k/3}}}{k^{3/4}}$ de Azevedo Pribitkin [2009] at the second equality, the maximal element is taken at the third, and the above bound of Robbins for factorial is used at the last. We comment that the tighter $O(MNL^2 1.56^L)$ can be seen to approximate the initial complexity accurately for $L \le 25$. cf., Table 2 in Section II.8.

The cumulative memory cost of tensor memorization is given by (8.3) (Proposition 29). By (8.4) at Proposition 30, for the computational complexity it suffices to calculate $\sum_{m=1}^{L}\sum_{b=0}^{m} C(b,m)$, where $C(b,m)$ is the complexity of $D^m_{\beta^b, \boldsymbol{x}^{m-b}}(Id - BA_\beta)$. We write,

$$\sum_{m=1}^{L}\sum_{b=0}^{m-1} C(b,m) = \sum_{m=1}^{L}\sum_{b=0}^{m-1}\left[O\left(\binom{m-b+M-1}{m-b}\binom{b+M-1}{b}\cdot M^2 N\right) + O\left(\binom{m-b+M-1}{m-b}(m-b)!M\right)\right] \quad \text{(E.16)}$$

Where, we have ignored the complexity $O(MN)$ of the L derivative tensors with respect to β alone, those with $b = m$, as they are negligible compared to the below. We comment that the first

summand in (E.16) pertains to the cost of computing mixed partial derivatives, while the second summand corresponds to copy operations; see Section III.E.2.3.

For the first summand at (E.16), by the formula $\sum_{k=0}^{r} \binom{r-k}{m}\binom{s+k}{n} = \binom{r+s+1}{m+n+1}$ [Knuth, 1997, Eq. (25) in 1.2.6], we have

$$\sum_{m=1}^{L}\sum_{b=0}^{m-1}\binom{m+M-1-b}{M-1}\binom{M-1+b}{M-1} \leq \sum_{m=1}^{L}\binom{m+2M-1}{2M-1}$$

$$\leq L\cdot\binom{2M+L-1}{2M-1} = L\cdot\frac{2M}{2M+L}\cdot\binom{2M+L}{2M}$$

$$\leq e^{(2M+L)H_e(\rho)+\frac{1}{2}\ln\frac{\rho L}{\pi}} \quad \text{(E.17)}$$

Where the first inequality follows by setting $r := m+M-1$ and $s := M-1$ in the above formula, summing only over $b = 0,\ldots,m-1$ rather than up to r. The second inequality follows by bounding with the largest summand $m = L$. For the last inequality, set $\rho := \frac{2M}{2M+L}$. It then follows from [Cover and Thomas, 2006, Lemma 17.5.1], up to straightforward modifications to measure entropy $H_e(\cdot)$ in nats instead of bits.

For the second summand at (E.16),

$$\sum_{m=1}^{L}\sum_{b=0}^{m-1}\binom{m-b+M-1}{m-b}(m-b)! = \frac{1}{(M-1)!}\sum_{m=1}^{L}\sum_{b=0}^{m-1}(m-b+M-1)!$$

$$\leq L^2\cdot\frac{M}{M+L}\frac{(M+L)!}{M!} \leq ML^2\cdot(M+L)^{L-1} = e^{(L-1)\ln(M+L)+\ln(ML^2)} \quad \text{(E.18)}$$

Where the inequalities follow by bounding $m-b$ by its largest value L, and $\frac{(M+L)!}{M!} = (M+1)\cdot(M+2)\cdots(M+L)$ by $(M+L)^L$.

Plugging (E.17) and (E.18) into (E.16) and then back into formula (8.4) for the computational complexity of root-tracking (Proposition 30), and adding the cost (E.15) of initial computational, we obtain

$$O\left(e^{(2M+L)H_e(\rho)+\frac{1}{2}\ln\frac{\rho L}{\pi}+\ln(M^2N)}\right) + O\left(e^{(L-1)\ln(M+L)+2\ln(ML)}\right)$$

$$+ O\left(e^{9/4\cdot\ln L+(L+1)\ln M+\pi\sqrt{2L/3}}\right) + O\left(e^{(L+5/2)\ln L-c_1 L+c_2+\ln MN}\right) \quad \text{(E.19)}$$

for the total computational cost when $L \geq 2$. These are bounded from above by,

$$O\left(e^{(2M+L)H_e(\rho)+(L+5/2)\ln L+(L+1)\ln M+(L-1)\ln(M+L)+\ln N+\pi\sqrt{2L/3}}\right)$$

$$< O\left(e^{(2M+L)H_e(\rho)+(3L+5/2)\ln(M+L)+\ln N+\pi\sqrt{2L/3}}\right)$$

$$= O\left(N\cdot(M+L)^{(3L+5/2)}e^{(2M+L)H_e(\rho)+\pi\sqrt{2L/3}}\right) \quad \text{(E.20)}$$

which completes the proof of (3.3) in Theorem 6.

When $L = 1$, one can see directly from (8.5) of Proposition 30 and from Table 2 in Section II.8 that the computational complexity in this case is

$$O\left(M^3\right) + O\left(MN\right) + O\left(M^3N\right) + O\left(M^2\right) = O\left(M^3N\right), \quad \text{(E.21)}$$

proving (3.4) at the Theorem.

For the total memory cost (3.5), we add the memory cost (E.14) of the initial computations to that needed for storing the derivative tensors, (8.3) in Proposition 29. This yields

$$O\left(L^{(L+5/2)}\ln L\right) + O\left(MNL^2\right) + O\left(M^L L\right) , \tag{E.22}$$

completing the proof. $\qquad\square$

F Binary Source with a Hamming distortion measure: an analytical solution

In this section, we develop the explicit equations of one of the simplest rate-distortion problems: a binary source with a Hamming distortion measure. As it admits an analytical solution, this problem can be used to verify the correctness of the theory and of its implementation. cf., Figure 2.3 for example. This problem was used throughout with parameters $p = 0.3$ and $2^{-1} \leq \beta \leq 2^5$, unless stated otherwise.

The *Hamming distortion* is defined by

$$d(x, \hat{x}) := \begin{cases} 0 & \text{if } x = \hat{x}, \\ 1 & \text{if } x \neq \hat{x} . \end{cases} \tag{F.1}$$

It is also called a *probability of error distortion*, as $\mathbb{E}[d(X, \hat{X})] = Pr(X \neq \hat{X})$. The proof of [Cover and Thomas, 2006, Theorem 10.3.1] shows that the achieving distribution of a binary source $X \sim$ Bernoulli(p), $p < 1/2$, with a Hamming distortion measure (F.1) is

$$Pr(\hat{X} = 1) = \frac{p - D}{1 - 2D} \tag{F.2}$$

when $D \leq p$, and $Pr(\hat{X} = 1) = 0$ otherwise (Eq. (10.21) there). The rate-distortion function for this problem is

$$R(D) = H(p) - H(D) \quad \text{for } 0 \leq D \leq \min\{p, 1 - p\} , \tag{F.3}$$

and zero-rate otherwise.

To exchange variables from D to β, we use the relation $R'(D) = -\beta$ [Berger, 1971, Theorem 2.5.1], when information is expressed in nats (logarithms taken in the natural basis). Plugging (F.3) in implies,

$$-\beta = \ln \frac{D}{1 - D} \quad \Longleftrightarrow \quad D = \frac{1}{1 + e^\beta} \tag{F.4}$$

for $\beta > 0$. Plugging this back into (F.2) yields an analytical solution in terms of β,

$$Pr(\hat{X} = 1) = \frac{1 - p \cdot (1 + e^\beta)}{1 - e^\beta} \tag{F.5}$$

This problem has a unique bifurcation, occurring when $Pr(\hat{X} = 1)$ first hits 0, at

$$\beta_c = \ln \frac{1 - p}{p} . \tag{F.6}$$

So long that $\beta \geq \beta_c$, the exact solution is given by (F.5), and is otherwise constant.

Having an analytical solution (F.5) in terms of β, one can easily evaluate its derivatives with respect to β of any order, at any point.

References

S. Agmon, E. Benger, O. Ordentlich, and N. Tishby. Critical slowing down near topological transitions in rate-distortion problems. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 2625–2630. IEEE, 2021.

C. O. Aguilar. An Introduction to Real Analysis. https://www.geneseo.edu/~aguilar/public/assets/courses/324/real-analysis-notes.pdf, 2021. [Online; accessed 14-October-2021].

G. E. Andrews. *The theory of partitions*. Number 2. Cambridge university press, 1998.

S. Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

K. Atkinson, W. Han, and D. E. Stewart. *Numerical solution of ordinary differential equations*, volume 108. John Wiley & Sons, 2011.

G. A. Baker Jr. Essentials of padé approximants. 1975.

T. Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, 1971.

R. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, 1972.

Y. Blau and T. Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *International Conference on Machine Learning*, pages 675–685. PMLR, 2019.

J. C. Butcher. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, 125(1-2):1–29, 2000.

J. C. Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 3rd edition, 2016.

G. Constantine and T. Savits. A multivariate faa di bruno formula with applications. *Transactions of the American Mathematical Society*, 348(2):503–520, 1996.

T. M. Cover and J. A. Thomas. *Elements of Information Theory, 2nd Edition*. John Wiley & Sons, 2006. ISBN 0471241954.

I. Csiszár. On the computation of rate-distortion functions (corresp.). *IEEE Transactions on Information Theory*, 20(1):122–124, 1974.

W. de Azevedo Pribitkin. Simple upper bounds for partition functions. *The Ramanujan Journal*, 18(1):113–119, 2009.

O. de Oliveira. The implicit and the inverse function theorems: easy proofs. *Real Analysis Exchange*, 39(1):207–218, 2014.

J. Dieudonné. *Foundations of modern analysis*. Academic Press, third printing, 1969.

D. S. Dummit and R. M. Foote. *Abstract algebra*. John Wiley & Sons, Inc., 3rd edition, 2004.

F. Faà di Bruno. Sullo sviluppo delle funzioni. *Annali di scienze matematiche e fisiche*, 6:479–480, 1855.

F. Faà di Bruno. Note sur une nouvelle formule de calcul différentiel. *Quarterly J. Pure Appl. Math*, 1(359-360):12, 1857.

R. Gilad-Bachrach, A. Navot, and N. Tishby. An information theoretic tradeoff between complexity and accuracy. *Learning Theory and Kernel*, 2003. URL http://link.springer.com/chapter/10.1007/978-3-540-45167-9_43.

S. Gottlieb. Lecture notes on "Euler's Method, Taylor Series Method, Runge Kutta Methods, Multi-Step Methods and Stability". http://www.cfm.brown.edu/people/sg/AM35odes.pdf, 2006.

M. Hardy. Combinatorics of partial derivatives. *The Electronic Journal of Combinatorics*, 13(R1): 1, 2006.

W. P. Johnson. The curious history of faà di bruno's formula. *The American mathematical monthly*, 109(3):217–234, 2002.

H. Kielhöfer. *Bifurcation Theory An Introduction with Applications to Partial Differential Equations*. Springer, 2nd edition, 2012. doi: 10.1007/978-1-4614-0502-3.

D. E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., USA, 1997. ISBN 0201896834.

Y. A. Kuznetsov. *Elements of applied bifurcation theory*, volume 112. Springer Science & Business Media, 3rd edition, 2004.

T.-W. Ma. Higher chain formula proved by combinatorics. *the electronic journal of combinatorics*, pages N21–N21, 2009.

G. Matz and P. Duhamel. Information geometric formulation and interpretation of accelerated blahut-arimoto-type algorithms. In *Information theory workshop*, pages 66–70. IEEE, 2004.

J. R. Munkres. *Topology*. Prentice Hall, second edition, 2000.

K. Nakagawa, Y. Takei, S.-i. Hara, and K. Watabe. Analysis of the convergence speed of the arimoto-blahut algorithm by the second-order recurrence formula. *IEEE Transactions on Information Theory*, 67(10):6810–6831, 2021.

J. M. Ortega. *Numerical analysis: a second course*. SIAM, 1990.

A. E. Parker, A. G. Dimitrov, and T. Gedeon. Symmetry breaking in soft clustering decoding of neural codes. *IEEE transactions on information theory*, 56(2):901–927, 2010.

H. Robbins. A remark on stirling's formula. *The American mathematical monthly*, 62(1):26–29, 1955.

S. Roman. The formula of faa di bruno. *The American Mathematical Monthly*, 87(10):805–809, 1980.

K. Rose. A mapping approach to rate-distortion computation and analysis. *IEEE Transactions on Information Theory*, 40(6):1939–1952, 1994.

K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

K. Rose, E. Gurewitz, and G. Fox. A deterministic annealing approach to clustering. *Pattern Recognition Letters*, 11:589–594, 1990. doi: 10.1016/0167-8655(90)90010-Y.

J. J. Rotman. *Advanced Modern Algebra, Part 2*, volume 180. American Mathematical Soc., 3rd edition, 2017.

J. Sayir. Iterating the arimoto-blahut algorithm for faster convergence. In *2000 IEEE International Symposium on Information Theory (Cat. No. 00CH37060)*, page 235. IEEE, 2000.

C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27 (3):379–423, 1948.

C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959.

C. R. Sims. Rate–distortion theory and human perception. *Cognition*, 152:181–198, 2016.

R. P. Stanley. Enumerative combinatorics, volume 1 second edition. *Cambridge studies in advanced mathematics*, 2011.

T. Sutter, D. Sutter, P. M. Esfahani, and J. Lygeros. Efficient approximation of channel capacities. *IEEE Transactions on Information Theory*, 61(4):1649–1666, 2015.

A. Thaheem and A. Laradji. Classroom note: A generalization of leibniz rule for higher derivatives. *International Journal of Mathematical Education in Science and Technology*, 34(6):905–907, 2003.

N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

E. W. Weisstein. Padé Approximant. URL https://mathworld.wolfram.com/PadeApproximant.html. Visited on April 3rd, 2022.

Y. Yu. Squeezing the Arimoto-Blahut algorithm for faster convergence. *IEEE Transactions on Information Theory*, 56(7):3149–3157, 2010.

S. Zemel. The combinatorics of higher derivatives of implicit functions. *Monatshefte für Mathematik*, 188(4):765–784, 2019.