Workshop on Material Appearance Modeling (2019)
H. Rushmeier and R. Klein Editors

Spectral Rendering with the Bounded MESE and sRGB Data

Christoph Peters[1], Sebastian Merzbach[2], Johannes Hanika[1,3] and Carsten Dachsbacher[1]

[1]Karlsruhe Institute of Technology
[2]University of Bonn
[3]Weta Digital

Abstract

In a recent journal paper, we introduced a technique to represent reflectance spectra by an arbitrary number of Fourier coefficients. As a special case, we converted tristimulus data to three Fourier coefficients. After summarizing this work, we introduce the Fourier sRGB color space. It is defined in terms of Fourier coefficients but designed to behave similar to sRGB. Textures stored in Fourier sRGB support efficient spectral rendering but can be compressed with techniques designed for sRGB textures. Compression errors are similar to sRGB.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

Color representation is fundamental in computer graphics. Tristimulus color spaces are sufficient to describe the perception of light by a standard observer. However, accurate rendering requires more information. Light transport results in products of emission spectra and reflectance spectra along the light path. Correct computation of the final tristimulus values requires evaluation of these products throughout the range of visible wavelengths.

Nonetheless, it is common practice to use RGB color spaces for textures storing surface albedos. Artists are familiar with them and the resulting textures can be stored compactly. There is even hardware acceleration for decompression of RGB textures on GPUs. Many renderers use the simplifying assumption that each channel corresponds to one wavelength and perform component-wise multiplication of RGB triples. This model is self-consistent, simple and highly efficient but fails to capture the more complex reality.

A more principled approach is to turn RGB triples into continuous reflectance spectra during rendering. This is accomplished by precomputed, piecewise linear fits [Smi99, OYH18] or moderately large lookup tables [Mac35, MSHD15]. Though, some of these techniques produce physically implausible spectra [Mac35, Smi99] while others suffer from discontinuous results [OYH18] or high bandwidth requirements [Mac35, MSHD15]. A more recent technique [JH19] guarantees a smooth and physically meaningful reflectance whenever this is possible and only uses a lookup table before rendering to convert the texture. However, the converted textures are more vulnerable to rounding errors such that compression becomes difficult.

In this paper, we extend a recent work of ours [PMHD19]. Our prior work proposes to represent reflectance spectra by a small number of Fourier coefficients. For the reconstruction of a continuous reflectance spectrum, we do not use a simple truncated Fourier series because its ringing could lead to a reflectance of less than zero or more than one. Instead, we derive the bounded maximum entropy spectral estimate (bounded MESE): A novel non-linear reconstruction that guarantees smooth results bounded between zero and one whenever this is possible at all. It is evaluated by means of a closed-form solution rather than a lookup table.

When full reflectance spectra are known, we convert them to a small number of Fourier coefficients, typically four to six. Reconstructions of real-world reflectance spectra from this compact representation turn out to be highly accurate. If full spectra are not available, data defined in tristimulus color spaces can be turned into Fourier coefficients using a lookup table before rendering. This approach is similar to prior work [JH19] but the Fourier coefficients are more robust to quantization errors because of their linear relationship to the original spectrum. We found that 10 bits per coefficient suffice.

After reviewing our recent work [PMHD19] in more detail (Section 2), we further explore possibilities for compression of Fourier coefficients generated from tristimulus data. We propose a novel color space called Fourier sRGB (Section 3). It relates to Fourier coefficients through a linear transform and a non-linearity applied per component (as in sRGB). By design, its three coefficients behave similarly to coefficients of sRGB. A lookup table maps each sRGB triple in $[0,1]^3$ to a matching Fourier sRGB triple in $[0,1]^3$. The resulting Fourier sRGB textures can be compressed in the same manner as common sRGB textures (Section 4). The impact of compression on color reproduction is similar to sRGB. Thus, spectral rendering without increased bandwidth requirements becomes viable.

2. The Bounded MESE and Spectral Rendering

Our bounded MESE [PMHD19] is derived as dual of the MESE [Bur75, PKHK15]. While the bounded MESE produces signals bounded between zero and one, the MESE only guarantees positive signals. Thus, the MESE is suitable for emission spectra while the bounded MESE offers meaningful reflectance spectra. We focus on reflectance since the need for compact storage is more pronounced for textures.

In this section, we summarize properties of the bounded MESE (Section 2.1), describe its application to spectral rendering (Section 2.2) and recapitulate the conversion from tristimulus color spaces to Fourier coefficients (Section 2.3). For a detailed derivation of the bounded MESE and the algorithm, we refer to our prior work [PMHD19].

2.1. The Bounded MESE

Consider a 2π-periodic, bounded signal $g(\varphi) \in [0,1]$. In Section 2.2, we describe a mapping from wavelengths λ to phases $\varphi \in \mathbb{R}$ such that $g(\varphi)$ can be thought of as reflectance spectrum. We represent this signal by the Fourier coefficients

$$c_j := \frac{1}{2\pi} \int_{-\pi}^{\pi} g(\varphi) \exp(-ij\varphi) \, \mathrm{d}\varphi \in \mathbb{C} \tag{1}$$

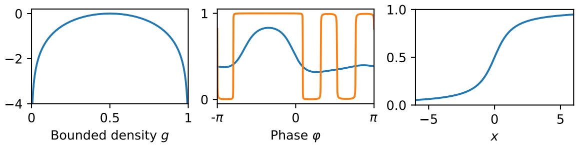
with $j \in \{0, \dots, m\}$.

While these $m+1$ Fourier coefficients are known exactly, infinitely many are unknown. A truncated Fourier series assumes that all unknown Fourier coefficients are zero. In doing so, it introduces ringing artifacts, which may lead to values below zero or above one.

The bounded MESE implements a drastically different approach. It constructs the unique signal $g(\varphi) \in [0,1]$ that satisfies Equation (1) and maximizes the log sin-entropy

$$\int_{-\pi}^{\pi} \log \sin(\pi g(\varphi)) \, \mathrm{d}\varphi.$$

This prior is visualized in Figure 1a. It rewards values near 0.5 while penalizing values near zero or one with scores going to minus infinity. In accordance with this entropy, the resulting reconstructions are smooth and avoid extreme reflectance values (Figure 1b, blue). Unlike a truncated Fourier series, they have energy throughout the spectrum. If the Fourier coefficients do not allow a smooth bounded reconstruction, the bounded MESE approaches reconstructions composed of box functions (Figure 1b, orange).



(a) log sin entropy (b) Bounded MESEs (c) $\frac{1}{\pi}\arctan x + \frac{1}{2}$

Figure 1: Due to the log sin entropy, the bounded MESE tends to avoid extreme values (1a). Still it can attain them, if needed (1b, orange). It is a Fourier series, mapped to $[0,1]$ by a scaled and shifted arctangent (1c).

Maximizing the log sin-entropy does not require a costly non-linear optimization. We have derived a closed-form solution for the bounded MESE. It takes time $O(m^2)$ overall and time $O(m)$ per phase φ where the bounded MESE needs to be evaluated. There are two variants of the algorithm. Which of the two is faster depends on how many samples of $g(\varphi)$ are evaluated. The algorithm that is faster for many samples constructs Lagrange multipliers $\lambda_0, \dots, \lambda_m$. Then the bounded MESE takes the form

$$g(\varphi) = \frac{1}{\pi} \arctan\left(\Re\lambda_0 + 2\Re \sum_{l=1}^{m} \lambda_l \exp(-il\varphi) \right) + \frac{1}{2}. \tag{2}$$

This representation also offers a compelling way to understand what kind of function the bounded MESE produces. It is a truncated Fourier series forced into the bounds between zero and one by a scaled and shifted arctangent (Figure 1c). Thanks to the way in which the Lagrange multipliers are computed, it matches the original Fourier coefficients c_j.

2.2. Mapping Wavelengths to Phases

From Equation (2), it is clear that the bounded MESE is a 2π-periodic function. Though, the reflectance spectra that we wish to describe are usually aperiodic. We overcome this problem using coefficients of a cosine transform. The phase for a wavelength $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ is defined as

$$\varphi = \pi \frac{\lambda - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} - \pi \in [-\pi, 0].$$

To define a signal $g(\varphi)$ on all of $[-\pi, \pi]$ we mirror it by setting $g(\varphi) := g(-\varphi)$ for all $\varphi \in [0, \pi]$. The moments $c_0, \dots, c_m$ for such an even signal are real. Thus, we only use half of our reconstruction but also need roughly half as much space to store the moments.

Additionally, it would be useful to account for the perceptual importance of particular wavelength ranges. We have to work with Fourier coefficients to be able to apply the

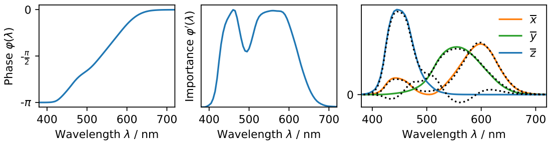


Figure 2: Our Fourier basis gets warped by $\varphi(\lambda)$ (left) and weighted by $\varphi'(\lambda)$ (middle). The warp is optimized for a good fit (dotted lines) to the color matching functions (right).

bounded MESE. Otherwise, a perceptually ideal choice for three coefficients would be CIE XYZ. While we cannot reach this goal exactly, we have one way to get close.

We optimize a monotonic function $\varphi(\lambda) \in [-\pi, 0]$ (Figure 2). Effectively, this mapping warps our Fourier basis and weights it by the derivative $\varphi'(\lambda)$. Our optimization maximizes similarity between the space spanned by this warped and weighted Fourier basis and the space spanned by the color matching functions for the CIE standard observer $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$. The optimized function $\varphi(\lambda)$ is stored as lookup table in 5 nm intervals. Its use is highly beneficial to the color reproduction of our approach, especially for a small number of Fourier coefficients.

2.3. Conversion of Tristimulus Data

Although our prior work focuses on a compact representation for spectral reflectance data, we acknowledge that the use of tristimulus color spaces is far more common. Therefore, we construct lookup tables mapping colors in tristimulus color spaces to three Fourier coefficients.

The construction is a two-step procedure. First we take a dense sampling of the space of Fourier coefficients, e.g. with 1024^3 samples. For each triple, we reconstruct a reflectance spectrum using the bounded MESE and compute CIE XYZ coefficients. In the second step, we enter all CIE XYZ triples into an index for nearest-neighbor queries. To have a meaningful notion of distances, we use CIE LAB. Then we use nearest-neighbor queries to fill a uniform grid in the relevant tristimulus color space with matching Fourier coefficients. For example, we create a grid with 256^3 samples to map sRGB triples to triples of Fourier coefficients.

3. Fourier Color Spaces

In this section, we introduce two new color spaces. Both of them are defined in terms of Fourier coefficients, not in terms of CIE XYZ. Thus, they do not determine perception of a stimulus by a standard observer uniquely but are compatible with the bounded MESE. Thanks to the warp described in Section 2.2, they behave similar to CIE XYZ nonetheless.

Our objective is more efficient compression of textures with three Fourier coefficients. Previously, we have demonstrated that 10 bits per Fourier coefficient are sufficient for good results. However, 30 bits per pixel are still a lot compared to a compressed sRGB texture. If all of our Fourier coefficients stem from a lookup table for sRGB, we would prefer a representation that can be compressed just like sRGB.

3.1. The Fourier XYZ Color Space

As an intermediate step, we define a linear transform mapping Fourier coefficients to a color space that resembles CIE XYZ. We call it Fourier XYZ. Due to the warp described in Section 2.2, our Fourier basis already spans a similar space. We solve three linear least squares problems to find three linear combinations of our Fourier basis that approximate the three color matching functions $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ optimally. Implicitly, we assume illuminant E here. If we take the same three linear combinations for our Fourier coefficients, we obtain three Fourier XYZ coefficients:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} := \begin{pmatrix} 107.105747 & 100.951852 & 69.1110083 \\ 106.856988 & 72.8910048 & -60.5204316 \\ 108.902261 & -172.572868 & 103.202221 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$

3.2. The Fourier sRGB Color Space

We use the method described in Section 2.3 to map 256^3 sRGB triples in $[0,1]^3$ to Fourier XYZ. Our method finds a valid reflectance spectrum for each sRGB triple. Figure 3a visualizes the range of the resulting Fourier XYZ coefficients using Fourier chromaticity

$$x_c := \frac{X_c}{X_c + Y_c + Z_c}, \qquad y_c := \frac{Y_c}{X_c + Y_c + Z_c}.$$

To define Fourier sRGB, we seek a triangle of minimal area that encloses this range in the Fourier chromaticity diagram. Once properly stated, this optimization problem is three-dimensional and a generic global minimizer solves it well.

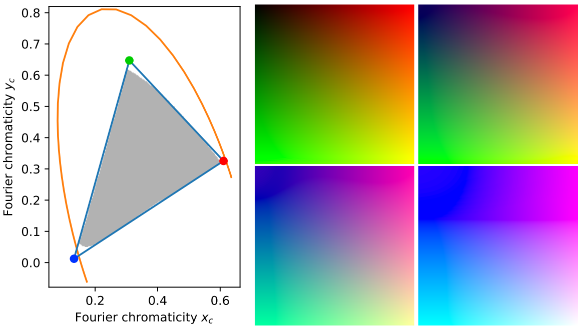
The three vertices of the resulting triangle should be thought of as primaries for red, green and blue. Except for the scaling, they define the three rows of a linear transform mapping Fourier XYZ to our linear Fourier sRGB. We determine the scaling of each row such that the maximal value across all sRGB samples is one. If we concatenate the resulting transform with the one from Fourier coefficients to Fourier XYZ, we obtain the following mapping:

$$\begin{pmatrix} R'_c \\ G'_c \\ B'_c \end{pmatrix} := \begin{pmatrix} .998343427 & 2.25573287 & 2.00536162 \\ .998381251 & 0.113152869 & -1.60073754 \\ .997450035 & -1.89517143 & 1.01411899 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}$$

We apply the non-linearity that converts rec. 709 into sRGB to each component to obtain the final Fourier sRGB coefficients R_c, G_c, B_c. With this color space, each sRGB triple corresponds to a Fourier sRGB triple in $[0,1]^3$.

3.3. Biasing

We can use a lookup table to map sRGB to Fourier sRGB. The bounded MESE serves to reconstruct the reflectance



(a) Chromaticity diagram (b) Slices at $B_c \in \{0, 0.33, 0.67, 1\}$

Figure 3: Visualizations of the Fourier sRGB color space. To the left we show how it encloses all samples needed to represent sRGB. The range covered by sRGB is grey, the spectral locus orange. To the right we show the sRGB colors for various slices through the Fourier sRGB color space.

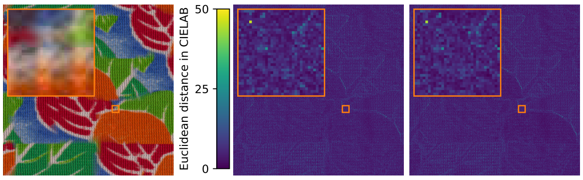
spectrum for each resulting Fourier sRGB triple. However, if we apply lossy compression directly to a Fourier sRGB texture, that may no longer be true. There are Fourier sRGB triples in $[0,1]^3$ for which no matching reflectance spectrum exists (see Section 3.3 in [PMHD19]).

We want to be able to treat compression as black box that may introduce arbitrarily large errors. Thus, we have to make a best effort to still provide a meaningful reflectance spectrum for Fourier sRGB triples corrupted in this manner. Our strategy is to manipulate Fourier coefficients during the reconstruction if they turn out to be invalid. Algorithm 2 projects them back into the valid domain efficiently. The details, which require an understanding of the inner workings of the reconstruction, are described in Appendix A.

4. Results

Figure 3b shows sRGB colors for a few slices of the Fourier sRGB color space. They resemble images where the x and y coordinates are mapped to R and G of sRGB. Hence, the overall behavior of Fourier sRGB is similar to sRGB, as intended. Colors change smoothly throughout the unit cube $[0,1]^3$. A small error in the Fourier sRGB coefficients R_c, G_c, B_c corresponds to a small change in color. The strongest non-linearities are visible in the slice for $B_c = 1$ because biasing is necessary for most of the upper third of this slice. Nonetheless, the behavior is smooth. Fourier sRGB triples in this part of the color space should only arise from compression artifacts such that smoothness is all we need.

To verify that compression of Fourier sRGB textures is unproblematic, we take a 1024^2 sRGB texture and convert it to Fourier sRGB. Then we compress the original texture and the Fourier sRGB texture to 210 kB using JPEG. To measure the perceptual error in the colors, we convert both compressed textures to the CIELAB color space and measure the Euclidean distance to ground truth per pixel. Figure 4 shows results. Although errors are distributed differently for sRGB and Fourier sRGB, the mean errors of 5.04 for sRGB and 5.26 for Fourier sRGB are close. The reconstructed Fourier sRGB texture (shown in Figure 4a using sRGB) only shows typical JPEG artifacts.



(a) Fourier sRGB (b) Error, Fourier sRGB (c) Error, sRGB

Figure 4: Results of JPEG compression applied to a Fourier sRGB texture (Figures 4a, 4b) and directly to the original sRGB texture (Figure 4c).

5. Conclusions

Our approach to spectral rendering used to have a disadvantage over sRGB when it comes to compact storage of textures. With the introduction of Fourier sRGB, we have eliminated this disadvantage. Existing compression methods are applicable and behave similar to sRGB. Thus, we offer principled spectral rendering without increased bandwidth requirements. It is a true alternative to component-wise multiplication of RGB triples and has many advantages over other techniques for reconstruction of reflectance from sRGB.

References

[Bur75] Burg J. P.: *Maximum Entropy Spectral Analysis*. Ph.D. dissertation, Stanford University, Department of Geophysics, 1975. URL: http://sepwww.stanford.edu/data/media/public/oldreports/sep06/. 2, 5

[JH19] Jakob W., Hanika J.: A low-dimensional function space for efficient spectral upsampling. *Computer Graphics Forum 38*, 2 (2019). doi:10.1111/cgf.13626. 1

[Mac35] MacAdam D. L.: Maximum visual efficiency of colored materials. *J. Opt. Soc. Am. 25*, 11 (1935). URL: http://www.osapublishing.org/abstract.cfm?URI=josa-25-11-361, doi:10.1364/JOSA.25.000361. 1

[MSHD15] Meng J., Simon F., Hanika J., Dachsbacher C.: Physically meaningful rendering using tristimulus colours. *Computer Graphics Forum 34*, 4 (2015). doi:10.1111/cgf.12676. 1

[OYH18] Otsu H., Yamamoto M., Hachisuka T.: Reproducing spectral reflectances from tristimulus colours. *Computer Graphics Forum 37*, 6 (2018), 370–381. doi:10.1111/cgf.13332. 1

[PKHK15] Peters C., Klein J., Hullin M. B., Klein R.:

Algorithm 1 Levinson's algorithm (see p. 14 ff. in [Bur75])
Input: Exponential moments $\gamma \in \mathbb{C}^{m+1}$.
Output: $q := C^{-1}(\gamma)e_0$.

```
$q_0^{(0)} := \frac{1}{\gamma_0}$
For $l \in \{1, \ldots, m\}$:
    $u^{(l)} := \sum_{k=0}^{l-1} q_k^{(l-1)} \gamma_{l-k}$
    $q^{(l)} := \frac{(q_0^{(l-1)}, \ldots, q_{l-1}^{(l-1)}, 0) - u^{(l)} (0, \overline{q_{l-1}^{(l-1)}}, \ldots, \overline{q_0^{(l-1)}})}{1 - |u^{(l)}|^2}$
Return $2\pi(q_0^{(m)}, \ldots, q_m^{(m)})$
```

Solving trigonometric moment problems for fast transient imaging. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 34*, 6 (2015). doi:10.1145/2816795.2818103. 2

[PMHD19] Peters C., Merzbach S., Hanika J., Dachsbacher C.: Using moments to represent bounded signals for spectral rendering. *ACM Trans. Graph. (Proc. SIGGRAPH) 38*, 4 (2019), 136:1–136:14. doi:10.1145/3306346.3322964. 1, 2, 4, 5

[Smi99] Smits B.: An rgb-to-spectrum conversion for reflectances. *Journal of Graphics Tools 4*, 4 (1999), 11–22. doi:10.1080/10867651.1999.10487511. 1

Appendix A: Biasing of Exponential Moments

The present appendix assumes familiarity with the derivations described in our prior work [PMHD19]. We use notations introduced there.

Evaluation of the bounded MESE always includes application of Levinson's algorithm (Algorithm 1) to solve a system of linear equations with the Toeplitz matrix $C(\gamma)$ and the canonical basis vector e_0. Levinson's algorithm solves this problem incrementally for growing main minors of the Toeplitz matrix. If we denote the $(l+1) \times (l+1)$ main minor of the Toeplitz matrix by $C_{(l)}(\gamma)$ for $l \in \{0, \ldots, m\}$,

$$C_{(l)}^{-1}(\gamma)e_0 = 2\pi q^{(l)}.$$

The bounded MESE is applicable for the bounded trigonometric moments $c \in \mathbb{C}^{m+1}$ if and only if $c_0 \in (0,1)$ and the Toeplitz matrix $C(\gamma)$ is positive definite. We fix a small constant $\varepsilon > 0$ (e.g. $\varepsilon = 10^{-4}$) and ensure $c_0 \in [\varepsilon, 1-\varepsilon]$ through simple clamping. If the Toeplitz matrix is positive definite, that also applies to all of its main minors $C_{(l)}(\gamma)$. In particular, we know for all $l \in \{0, \ldots, m\}$

$$q_0^{(l)} = \frac{1}{2\pi} e_0^* C_{(l)}^{-1}(\gamma) e_0 > 0.$$

This criterion is not only necessary but also sufficient because by Cramer's rule

$$e_0^* C_{(l)}^{-1}(\gamma) e_0 = \frac{\det C_{(l-1)}(\gamma)}{\det C_{(l)}(\gamma)}$$

where $\det C_{(-1)}(\gamma) := 1$.

Algorithm 2 Levinson's algorithm with biasing.
Input: Exponential moments $\gamma \in \mathbb{C}^{m+1}$ and a bias $\varepsilon > 0$.
Output: $\gamma \in \mathbb{C}^{m+1}$ such that $C(\gamma)$ is positive definite and $q := C^{-1}(\gamma)e_0$.

```
$q_0 := \frac{1}{\gamma_0}$
For $l \in \{1, \ldots, m\}$:
    $u := \sum_{k=0}^{l-1} q_k \gamma_{l-k}$
    If $|u| \geq 1$:
        $u := (1-\varepsilon) \frac{u}{|u|}$
        $\gamma_l := \frac{1}{q_0} \left( u - \sum_{k=1}^{l-1} q_k \gamma_{l-k} \right)$
        $\varepsilon := 1$
    $(q_0, \ldots, q_l) := \frac{(q_0, \ldots, q_{l-1}, 0) - u(0, \overline{q_{l-1}}, \ldots, \overline{q_0})}{1 - |u|^2}$
Return $\gamma$, $2\pi(q_0, \ldots, q_m)$
```

Since

$$q_0^{(l)} = \frac{q_0^{(l-1)}}{1 - |u^{(l)}|^2},$$

we find that the Toeplitz matrix $C(\gamma)$ is positive definite if and only if $|u^{(l)}| < 1$ for all $l \in \{0, \ldots, m\}$. With this insight, we are prepared to introduce our biasing strategy.

If $|u^{(l)}| \geq 1$, we replace $u^{(l)}$ by

$$(1-\varepsilon) \frac{u^{(l)}}{|u^{(l)}|}.$$

Then we propagate this change back to γ_l:

$$u^{(l)} = q_0^{(l-1)} \gamma_l + \sum_{k=1}^{l-1} q_k^{(l-1)} \gamma_{l-k}$$

$$\Rightarrow \gamma_l = \frac{1}{q_0^{(l-1)}} \left(u^{(l)} - \sum_{k=1}^{l-1} q_k^{(l-1)} \gamma_{l-k} \right)$$

Intuitively, the inequality $|u^{(l)}| < 1$ forces γ_l to reside in a disk of radius $\frac{1}{q_0^{(l-1)}}$ and with center

$$-\frac{1}{q_0^{(l-1)}} \sum_{k=1}^{l-1} q_k^{(l-1)} \gamma_{l-k}.$$

Our biasing strategy simply pulls it back to the closest point within this disk. Once γ_l is biased in this manner, the minor $C_{(l)}(\gamma)$ is positive definite but nearly singular. All higher order moments (if any) are almost uniquely determined. For reasons of numerical stability, it is beneficial to apply maximal biasing to these higher-order moments, i.e. to set $\varepsilon = 1$ after its first use. Algorithm 2 implements our strategy. It omits iteration indices in identifiers and a few common subexpressions have not been eliminated for readability.