Response regarding `TOMS-2020-0061`

Dr. Tim Hopkins,

After carefully reading and considering the comments from the reviewer, we would like to request additional reviews. Addressed individually below, we fear many of the comments made by the reviewer do not appreciably assess this work, nor its relevance to the community. Each numbered item and italicized text block below is a comment provided by the reviewer. Our responses immediately follow.

(1) *A very similar algorithm for monotone quintic spline interpolation has already been published and detailed in pseudo-code:*

> *Thomas C.H. Lux, Layne T. Watson, et al: An Algorithm for Constructing Monotone Quintic Interpolating Splines. Proceedings of the 2020 Spring Simulation Conference, May 2020, Article No. 33, Pages 1-12.*

> *The differences with respect to the current manuscript seem to be rather specific and minor, and at least I would have expected a discussion/comparison between the current approach and the one in the above published work.*

That conference paper was an earlier version of this work and may appear similar at first glance, but there are many critical differences between the two (computations refactored for numerical stability, different method for estimating derivatives, using a binary search to find feasible derivative values for the entire spline rather than a single interval). Notably, there was no code published with that conference paper, which is one of the most important components of this submission. To make these differences more clear, two sentences were added to the end of paragraph 4 in section 1.

(2) *The overall procedure for computing a monotone spline interpolant is rather basic and quite standard. The only special ingredient is the use of a binary search in the last step, but the same idea has also been used in the paper mentioned in (1).*

Whereas the overall procedure for computing a monotone spline is rather basic and standard for splines of lower order than quintic, there is no published algorithm with software (that we are aware of) that creates a piecewise quintic monotone spline interpolant using theoretically tight constraints. The submitted work resembles the methodology for constructing lower order spline interpolants because there are similar theoretical constructs. However, this work is distinctly different and more capable because the resulting monotone spline is quintic (twice continuously differentiable). A quick look at the theory involved in checking quintic monotonicity (cf. Ulrich and Watson) shows it is significantly more complex than the cubic theory, which also explains why no high quality quintic code has been produced before.

The binary search referred to in (1) is used to make a single interval of the interpolating spline monotone. That is distinctly different than the binary search in the present submission that searches for derivative values where *all* intervals are monotone. The only similarity between the papers in that regard is that they both utilize a binary search at some point.

(3) *The literature study on monotone spline interpolation is incomplete and misses important work. As far as I am aware of, one of the most commonly used methods for shape-preserving interpolation is the quadratic spline method by Larry Schumaker (it does not only preserves monotonicity but also convexity!), at least in the spline approximation theory community:*

> *Larry L. Schumaker: On Shape Preserving Quadratic Spline Interpolation. SIAM Journal on Numerical Analysis, Vol. 20, No. 4 (1983), Pages 854-864.*

The submission cites quadratic, cubic, and quartic previous works. The work by Larry Schumaker is now referenced in the attached update to the paper (section 1, paragraph 3). It is our understanding that one of

the most widely used shape-preserving spline algorithms is a cubic spline methodology proposed by Fritsch and Carlson (known as PCHIP), which is included in many mathematical software libraries and cited in the submission.

(4) *The numerical section just shows few examples of data sets and the obtained spline results. The authors make some general statements about the algorithm like "it generally provides aesthetically pleasing (low wiggle) initial estimates … is preferred because it results in monotone quintic spline interpolants that are nearer to initial estimates" but a deeper (more objective) discussion is lacking. Also a numerical comparison with other classical monotone spline methods is missing, at least the algorithm by Schumaker [1983] and also the one by Fritz/Carlson [1980]. This might help understand why the new method is of interest.*

This is a reasonable criticism, but also easily addressed with minor revisions to the submission. While the reviewer may disagree with the language we used, there are objective reasons for the decisions made in this work in addition to the subjective comments made. Specifically, the quadratic facet model picks the minimum curvature local estimate for second derivatives which has the smallest magnitude second derivative, and the binary search identifies derivative values absolutely nearer to the initial estimates than the previous method of shrinking derivative values used in (1). These facts are now mentioned in the paper (section 4, paragraphs 2 and 3).

Comparisons with lower order methods are not appropriate for this work. There is no published mathematical software that also constructs a monotone quintic spline interpolant to data. If a user desires a monotonicity preserving C^2 approximation, this would be the only published software that achieves that goal. This paper is not intended to be a review and comparison of all available shape preserving spline packages. Including such comparisons would also require subjective decisions about what visuals to show, and those decisions might draw criticisms that distract from the unique purpose and achievement of this work (producing C^2 monotonicity preserving approximations to data). To emphasize that this work is not a replacement for previous works, but rather an additional method to consider, the first sentence of section 1, paragraph 5 that mentioned the importance of this work "over" previous methods was removed. The inherent difficulty of comparing this work with existing lower order methods is now mentioned in section 5, paragraph 5.

(5) *The most time-consuming part of the algorithm is the conversion to the B-spline representation. This has been done by solving a global linear system. It would be more efficient when using a local quasi-interpolation spline scheme with projection property.*

Moreover, spline evaluation by explicitly computing the linear combination of B-splines is not efficient; instead use de Boor-Cox algorithm.

The reviewer may have missed the reason for the global linear system, from the submission (section 5, paragraph 4), "It would be faster to construct splines over intervals independently ... the single linear system is chosen here for the decreased redundancy". Comments in the paper and the Fortran source code indicated an option for returning the derivatives at the data points, so one can use either the piecewise polynomial or the B-spline representations. Further, the time for the banded system solve is $O(n)$, reasonable for n data points. To be clear, the de Boor-Cox algorithm *is* used to evaluate the underlying B-splines (section 5, paragraph 1). The linear combination in the submission is of the (quintic) B-splines that define the interpolant.

To the best of our knowledge, rejection based on a single review has not been a journal policy and we respectfully request additional reviews. We believe this algorithm and the accompanying code has great potential for TOMS, noting that the quadratic and cubic works preceding this have remained state-of-the-art for forty years. Thank you for both your time and consideration.

Sincerely,

Thomas Lux, on behalf of all authors.