AN ALGORITHM FOR CONSTRUCTING MONOTONE PIECEWISE QUINTIC HERMITE INTERPOLATING POLYNOMIALS

Thomas C. H. Lux
Layne T. Watson
Tyler H. Chang

Department of Computer Science
Virginia Polytechnic Institute and State University
2000 Torgersen Hall, Blacksburg, VA, USA
{tchlux,ltwatson,thchang}@vt.edu

Li Xu
Yueyao Wang
Yili Hong

Department of Statistics
Virginia Polytechnic Institute and State University
213 Hutcheson Hall, Blacksburg, VA, USA
yilihong@vt.edu

ABSTRACT

An algorithm for computing monotone piecewise quintic interpolating polynomials is proposed. Algebraic constraints for enforcing monotonicity are provided that align with quintic monotonicity theory. The algorithm is implemented, tested, and applied to several sample problems to demonstrate the improved accuracy of piecewise quintic monotone interpolation compared to the existing piecewise cubic solution.

Keywords: monotone, spline, Hermite interpolation, quintic polynomial.

1 INTRODUCTION AND MOTIVATION

Many domains of science rely on smooth approximations to real-valued functions over a closed interval. Piecewise polynomial functions (splines) provide the smooth approximations for animation in graphics (Herman and Oftedal 2015, Quint 2003), aesthetic structural support in architecture, efficient aerodynamic surfaces in automotive and aerospace engineering (Brennan 2019), prolonged effective operation of electric motors (Berglund, Brodnik, Jonsson, Staffanson, and Soderkvist 2009), and accurate non parametric approximations in statistics (Knott 2012). While polynomial interpolants or regressors apply broadly, splines are often a good choice because they can approximate globally complex functions while minimizing the local complexity of an approximation.

It is often the case that the true underlying function or phenomenon being modeled has known properties e.g., convexity, positivity, various levels of continuity, or monotonicity. Given a reasonably large amount of data, it can be impossible to maintain these properties with a single polynomial function. In general, the maintenance of function properties through interpolation / regression is usually referred to as *shape preserving* (Fritsch and Carlson 1980, Gregory 1985). The specific shapes this work will focus on are monotonicity and multiple levels of continuity for a function. These properties are chiefly important to the approximation of cumulative distribution functions and subsequently the effective generation of random numbers from a specified distribution.

In statistics especially, the construction of a monotone interpolating spline that is continuous in second derivative is meaningfully useful (Ramsay et al. 1988). A function with these properties could approximate random variables to a high level of accuracy with relatively few intervals. A continuously twice differ-

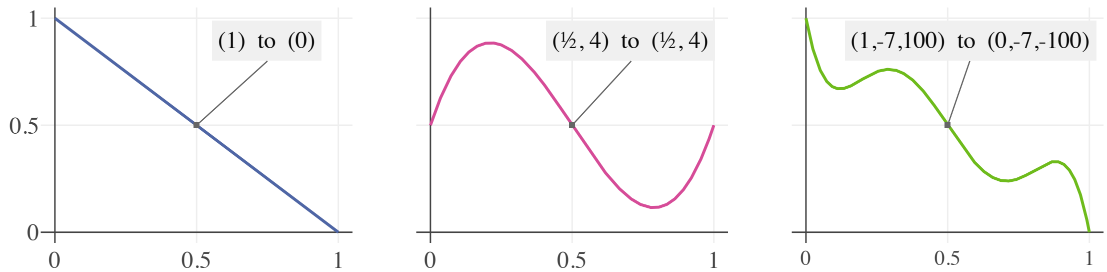

Figure 1: Example polynomials that interpolate function values at the ends of the interval $[0,1]$. The first only interpolates the function values $f(0) = 1$ and $f(1) = 0$, making it the order two polynomial $f(x) = 1-x$. For the second plot $f(x) = 8x^3 - 12x^2 + 4x + 1/2$, which is order four and interpolates the values $f(0) = 1/2$, $f'(0) = 4$, $f(1) = 1/2$, $f'(1) = 4$. Finally the third plot shows the order six polynomial $f(x) = -64x^5 + 160x^4 - 140x^3 + 50x^2 - 7x + 1$ interpolating the function values $f(0) = 1$, $f'(0) = -7$, $f''(0) = 100$, $f(1) = 0$, $f'(1) = -7$, $f''(1) = -100$. Notice that interpolating the same fixed number of function values at each endpoint will always result in an even order interpolating polynomial.

entiable approximation to a cumulative distribution function (CDF) would also produce a corresponding probability density function (PDF) that is continuously differentiable, which is a property many standard parametric distributions maintain.

The currently available software for monotone piecewise polynomial interpolation includes quadratic (He and Shi 1998), cubic (Fritsch and Carlson 1980), and quartic (with limited application) (Wang and Tan 2004, Piah and Unsworth 2011, Yao and Nelson 2018) cases. Even though theory has been provided for the quintic case (Ulrich and Watson 1994, Hess and Schmidt 1994), this theory has not yet been used to construct a piecewise quintic monotone spline interpolation routine. Recent work suggests that the lack of quintic software may be due to a general unawareness of the theory (Xie, King, Hong, and Yang 2018).

The importance of piecewise quintic interpolation over lower order approximations can be simply demonstrated. In general, the order of a polynomial determines the number of function values it can interpolate, and the growth rate of error away from the interpolated function values. As demonstrated in Figure 1, it can be seen that matching a value at either end of the interval requires an order two (linear) approximation and each additional derivative at the ends of the interval raises the necessary polynomial order by two. The body of this work is composed of a novel algorithm for enforcing monotonicity on quintic polynomial pieces, then extending that solution to work on generic splines.

The major contribution of this work is an algorithm for constructing piecewise quintic Hermite interpolating polynomials that utilizes existing quintic monotonicity theory. The remainder of this paper is structured as follows: ...

1.1 Computing a Monotone Cubic Interpolant

The current state of the art monotone interpolating spline with a mathematical software implementation is piecewise cubic, continuously differentiable, and was first proposed in (Fritsch and Carlson 1980) then expanded upon in (Carlson and Fritsch 1985). Let $\pi : x_0 = k_1 < k_2 < \cdots < k_n = x_1$ be a partition of the interval $[x_0, x_1]$. Let $f : \mathbb{R} \to \mathbb{R}$ and $\{f(k_i) : i = 1, 2, \ldots, n\}$ be a given set of data values at the partition points for a monotone function f, meaning $f(k_i) \le f(k_{i+1})$ for $i = 1, \ldots, n-1$ or $f(k_i) \ge f(k_{i+1})$ for $i = 1, \ldots, n-1$. Let $\hat{f}$ be a piecewise cubic spline defined in each sub-interval $I_i = [k_i, k_{i+1}]$ by

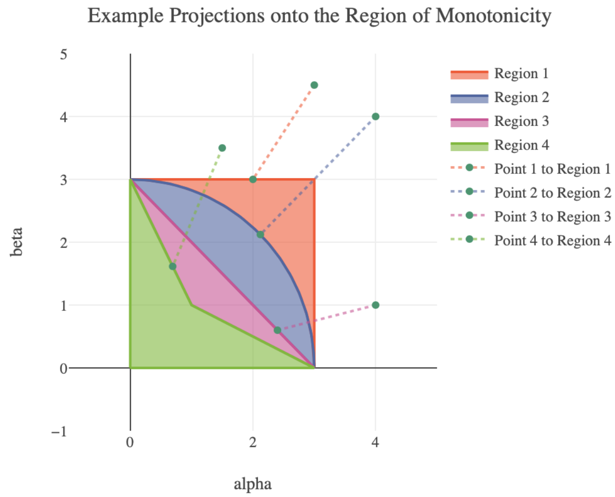

Figure 2: These are the feasible regions of monotonicity for cubic splines and the projections that make a cubic polynomial piece monotone.

$$\begin{aligned}
h_i &= k_{i+1} - k_i \\
u(t) &= 3t^2 - 2t^3 \\
p(t) &= t^3 - t^2 \\
\hat{f}(x) &= f(k_i)\, u\big((k_{i+1} - x)/h_i\big) + f(k_{i+1})\, u\big((x - k_i)/h_i\big) \\
&\quad - \hat{f}\,'(k_i)\, p\big((k_{i+1} - x)/h_i\big) + \hat{f}\,'(k_{i+1})\, p\big((x - k_i)/h_i\big).
\end{aligned}$$

Notice that a trivially monotone spline results when $\hat{f}\,'(k_i) = 0$, for $i = 1, \ldots, n$. However, such a spline has too many *wiggles* for most applications. Fritsch and Carlson show that simple conditions on the derivative values can guarantee monotonicity, and that these conditions can be enforced in a way that ensures modifications on one interval will not break the monotonicity of cubic polynomials over any neighboring intervals. Consider the terms $\alpha = \big(\hat{f}\,'(k_i)(k_{i+1} - k_i)\big)/\big(f(k_{i+1}) - f(k_i)\big)$ and $\beta = \big(\hat{f}\,'(k_{i+1})(k_{i+1} - k_i)\big)/\big(f(k_{i+1}) - f(k_i)\big)$, now monotonicity of a cubic polynomial over a sub-interval can be maintained by ensuring that α and β reside in any of the regions depicted in Figure 2.

The actual region of monotonicity for a cubic polynomial is larger, but projection of (α, β) into one of these regions ensures that monotonicity will be achieved and not violated for neighboring regions. The user must decide which region is most for the projections based on the application, Fritsch and Carlson recommend using region 2.

While the cubic monotonicity case presents a straight-forward solution, the region of monotonicity is not so simple in the quintic case. In the next section, an algorithm for performing a projection similar to those for cubic polynomials is proposed.

2 MONOTONE QUINTIC INTERPOLATION

When provided data that has no assigned first and second derivative values, the derivative data is filled by a linear fit of neighboring data points. End points are set to be the slope between the end and its nearest neighbor.

The method for finding the transition point of a boolean function on a line is the Golden Section search. This will be referred to in pseudo code as `line_search(`g`, `a`, `b`)` where $a, b \in S$ for S closed under convex combination, $g : S \to \{0,1\}$ is a boolean function, and $g(b) = 1$. If $g(a) = 1$ then a is returned, otherwise the smallest $c \in [0,1]$ such that $g\big(a(1-c)+cb\big) = 1$ is returned.

After assigning function values and derivative values, an interpolating function is constructed from a quintic B-spline basis.

2.1 Verifying Monotonicity of a Quintic Polynomial

Let f be a quintic polynomial over a closed interval $[x_0, x_1] \subset \mathbb{R}$. Now f is uniquely defined by the evaluation tuples $\big(x_0,\ f(x_0),\ f'(x_0),\ f''(x_0)\big)$ and $\big(x_1,\ f(x_1),\ f'(x_1),\ f''(x_1)\big)$. Assume without loss of generality that $f(x_0) < f(x_1)$, where the case of monotonic decreasing f would consider the negated the function values. The following algorithm will determine whether or not f is monotone increasing on the interval $[x_0, x_1]$.

Algorithm 1a: `is_monotone`

0: `if` $\big(f'(x_0) = 0$ `or` $f'(x_1) = 0\big)$ `return is_monotone_simplified`

1: `if` $\big(f'(x_0) < 0$ `or` $f'(x_1) < 0\big)$ `return FALSE`

This can be seen clearly from the fact that f is analytic; there will exist some nonempty interval about x_0 or x_1 for which f' is negative.

2: $A = f'(x_0)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$

3: $B = f'(x_1)\frac{x_1 - x_0}{f(x_1) - f(x_0)}$

The variables A and B correspond directly to the theoretical foundation for positive quartic polynomials established in (Ulrich and Watson 1994), first defined after equation 18.

8: $\gamma_0 = 4\frac{f'(x_0)}{f'(x_1)}(B/A)^{3/4}$

9: $\gamma_1 = \frac{x_1 - x_0}{f'(x_1)}(B/A)^{3/4}$

4: $\alpha_0 = 4(B/A)^{1/4}$

5: $\alpha_1 = -\frac{x_1 - x_0}{f'(x_1)}(B/A)^{1/4}$

6: $\beta_0 = 30 - \frac{12\big(f'(x_0)+f'(x_1)\big)(x_1 - x_0)}{\big(f(x_1)-f(x_0)\big)\sqrt{A}\sqrt{B}}$

7: $\beta_1 = \frac{-3(x_1 - x_0)^2}{2\big(f(x_1)-f(x_0)\big)\sqrt{A}\sqrt{B}}$

The γ, α, and β terms with subscripts 0 and 1 are algebraic reductions of the simplified conditions for satisfying Theorem 2 in (Ulrich and Watson 1994) (equation 16). These terms with subscripts 0 and 1 give the computation of α, β, and γ the form seen in lines 10-12 below.

10: $\gamma = \gamma_0 + \gamma_1 f''(x_0)$

11: $\alpha = \alpha_0 + \alpha_1 f''(x_1)$

12: $\beta = \beta_0 + \beta_1\big(f''(x_0) - f''(x_1)\big)$

13: `if` $(\beta \le 6)$ `then return` $\alpha > -(\beta+2)/2$

else return $\gamma > -2\sqrt{\beta - 2}$

Given the same initial conditions there are special circumstances which allow for the usage of simpler monotonicity conditions. In this case, consider when the quintic function has either $f'(x_0) = 0$ or $f'(x_1) = 0$. This reduces the problem of verifying monotonicity to one of cubics established by (**?**).

Algorithm 1b: `is_monotone_simplified`

$\alpha = 30 - \frac{(x_1-x_0)\left(14f'(x_0)+16f'(x_1)-\left(f''(x_1)-f''(x_0)\right)(x_1-x_0)\right)}{2\left(f(x_1)-f(x_0)\right)}$

$\beta = 30 - \frac{(x_1-x_0)\left(2f'(x_0)+24f'(x_1)-\left(f''(x_0)+3f''(x_1)\right)(x_1-x_0)\right)}{2\left(f(x_1)-f(x_0)\right)}$

$\gamma = \frac{(x_1-x_0)\left(7f'(x_0)+f''(x_0)(x_1-x_0)\right)}{f(x_1)-f(x_0)}$

$\delta = \frac{f'(x_0)(x_1-x_0)}{f(x_1)-f(x_0)}$

*The variables above are algebraic expansions of the coefficients for the cubic derivative function in (**?**).*

if $\left(\mathtt{min}(\alpha,\delta) < 0\right)$ return FALSE

else if $\left(\beta < \alpha - 2\sqrt{\alpha\delta}\right)$ return FALSE

else if $\left(\gamma < \delta - 2\sqrt{\alpha\delta}\right)$ return FALSE

else return TRUE

Next the modification of a quintic spline to enforce monotonicity will be discussed.

2.2 Enforcing Monotonicity of a Quintic Polynomial

Algorithm 2a: `make_monotone`

---

if $\left(f(x_1) - f(x_0) = 0\right)$ return $f'(x_0) = f'(x_1) = f''(x_0) = f''(x_1) = 0$

$f'(x_0) = \mathtt{median}\left(0, f'(x_0), 14\frac{f(x_1)-f(x_0)}{x_1-x_0}\right)$

$f'(x_1) = \mathtt{median}\left(0, f'(x_1), 14\frac{f(x_1)-f(x_0)}{x_1-x_0}\right)$

*This selection of values for $f'(x_0)$ and $f'(x_1)$ is suggested by (Ulrich and Watson 1994) (originally from (**?**)), and quickly enforces upper and lower bounds on derivative values to ensure quintic monotonicity is obtainable.*

$A = f'(x_0)\frac{x_1-x_0}{f(x_1)-f(x_0)}$

$B = f'(x_1)\frac{x_1-x_0}{f(x_1)-f(x_0)}$

if $AB \leq 0$ return `make_monotone_simplified`

if $\left(\mathtt{max}(A,B) > 6\right)$

$f'(x_0) = 6f'(x_0)\ /\ \mathtt{max}(A,B)$

$f'(x_1) = 6f'(x_1)\ /\ \mathtt{max}(A,B)$

This box bound ensures that (A,B) remains within a viable region of monotonicity (satisfying Theorem 4, seen in Fig. 6 of (Ulrich and Watson 1994)).

7: $\hat{f}''(x_0) = -\sqrt{A}\big(7\sqrt{A}+3\sqrt{B}\big)\frac{f(x_1)-f(x_0)}{(x_1-x_0)^2}$

$\hat{f}''(x_1) = \sqrt{B}\big(3\sqrt{A}+7\sqrt{B}\big)\frac{f(x_1)-f(x_0)}{(x_1-x_0)^2}$

This selection of values of f'' is guaranteed to satisfy Theorem 4 from (Ulrich and Watson 1994) and is chosen because it is (reasonably) the average of the two endpoints of the interval of monotonicity for second derivative values.

8: $\eta = \big(f''(x_0), f''(x_1)\big)$

$\eta_0 = \big(\hat{f}''(x_0), f''(x_1)\big)$

$f''(x_0), f''(x_1) =$ `line_search`$\big($`is_monotone`, η, $\eta_0\big)$

Similar to the simplified check for monotonicity, when the derivative value at one endpoint of the interval is 0, a simplified set of steps can be taken to enforce monotonicity.

Algorithm 2b: `make_monotone_simplified`

0: $f''(x_0) = \texttt{max}\left(f''(x_0), \frac{-6f'(x_0)}{x_1-x_0}\right)$

*Considering the α, γ, β, and δ defined in (**?**), this first step enforces $\gamma > \delta$. It is already guaranteed that $\delta = \frac{f'(x_0)(x_1-x_0)}{f(x_1)-f(x_0)} > 0$. Only two conditions remain to guarantee monotonicity.*

1: $f''(x_1) = \texttt{max}\left(f''(x_1), f''(x_0) + \frac{14f'(x_0)+16f'(x_1)+60(f(x_1)-f(x_0))/(x_1-x_0)}{x_1-x_0}\right)$

Now it is guaranteed that $\alpha \geq 0$.

2: $f''(x_1) = \texttt{max}\left(f''(x_1), 6f'(x_0) - 4f'(x_1) - f''(x_0)\right)$

*Lastly, this guarantees that $\beta \geq \alpha$. All conditions are met to satisfy proposition 2 of (**?**) and ensure monotonicity.*

Notice that all above algorithms (assuming a fixed level of precision is desired) have $\mathscr{O}(1)$ runtime. Only a finite number of operations are needed for monotonicity verification. A line search is performed for enforcement, however that search requires a fixed number of steps to achieve any predetermined relative precision on the line.

2.3 Constructing a Piecewise Quintic Monotone Spline

Finally, the construction of a full piecewise quintic spline is outlined using the above algorithms. Let $f : \mathbb{R} \to \mathbb{R}$ be a function in $\mathscr{C}^2$. Proceed given evaluation tuples $\big(x_i,\, f(x_i),\, f'(x_i),\, f''(x_i)\big)$ for $i = 0,\ldots,N$ such that $x_i < x_{i+1}$ and (without loss of generality) $f(x_i) \leq f(x_{i+1})$ for $i = 1,\, \ldots,\, N-1$.

Algorithm 3: `monotone_spline`

```
for i = 0,...,N-1
  if (not is_monotone(i,i+1)) make_monotone(i,i+1)
```

In the shorthand notation above, i and $i+1$ refer to the associated tuples of the form $(x_i,\ f(x_i),\ f'(x_i),\ f''(x_i))$. This notation will be used through the remainder of this algorithm.

```
  for j = i-1,...,0
    if (not is_monotone(j,j+1)) make_monotone(j,j+1)
    else break
```

The above 'for' loop will be referred to as the cascade fix. If the adjustment of second derivative values causes the previous interval to become nonmonotone, then the left-hand second derivative value must be updated. This may (abnormally) require adjustments across all previously visited intervals.

```
  while (not is_monotone(i,i+1))
    f'(x_i) = f'(x_i)/k
    f'(x_{i+1}) = f'(x_{i+1})/k
    make_monotone(i,i+1)
```

In the case that the two corrections to neighboring intervals contradict, the first derivative values of the active interval are decreased to enlarge the overlap of regions I and II (from (Ulrich and Watson 1994)) of the two intervals.

```
    for j = i-1,...,0
      if (not is_monotone(j,j+1)) make_monotone(j,j+1)
      else break
```

Finally an additional 'cascade fix' is performed to ensure that all previous intervals are still monotone after shrinking the derivative values of the current interval.

It is mentioned in (Ulrich and Watson 1994) that for sufficiently small $f'(x_i)$ and $f'(x_{i+1})$ the admissible solution interval of second derivative values becomes arbitrarily large. It can also be seen that decreasing the assigned derivative on right-hand side of an interval always allows the achievement of monotonicity because shrinking $f'(x_1)$ results in γ growing faster than $\sqrt{\beta}$ for *algorithm 1a*, for *algorithm 1b* β and γ will grow faster α and δ respectively. In application, one needs only to pick some $k > 1$ to ensure successful termination.

Given the potential for successive cascade fixes, the runtime of *algorithm 3* is $\mathcal{O}(N^2)$. In practice this has been observed to be incredibly unlikely and difficult to produce. While single cascade fixes can occur from hand crafted examples, the generously rounded values selected to ensure monotonicity frequently prevent successive cascade fixes.

3 EXPERIMENTAL RESULTS

- show a Figure of a simple quintic interpolant and its derivatives

3.1 Approximating a Trigonometric Function

For this experiment, the function $\sin(x) + x$ is considered over the interval $[0, (5/2)\pi]$.

- Pick a trig function, approximate it with cubic and quintic

- Show a Figure, and a Table of the maximum error and distribution of error across the segment

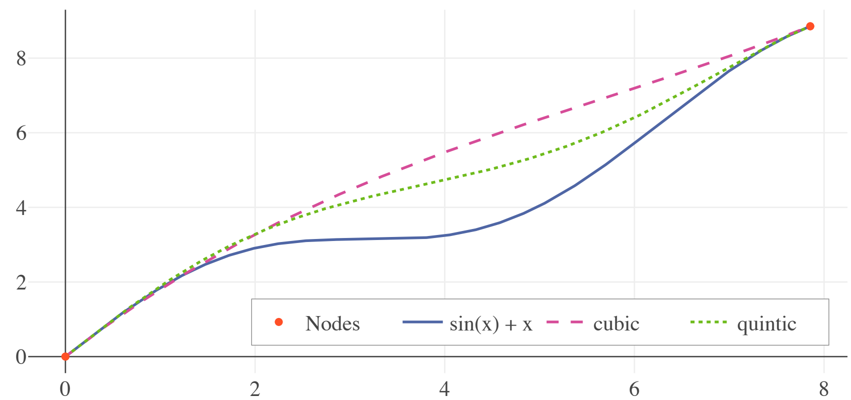

Figure 3: Depicted above are the monotone cubic and quintic spline interpolant of the function $\sin(x)+x$ over the nodes $[0,(5/2)\pi]$. Notice that the maximum error of the cubic interpolant is larger, because it only captures first derivative information at the nodes. The quintic interpolant captures both first and second derivative information at the nodes.

3.2 Approximating a Cumulative Distribution Function

- generate large sets of random monotone data, show Table of number of metrics growing with increasing number of nodes

- show visual of the *error*

3.3 Random Monotone Data

- generate large sets of random monotone data, show Table of number of metrics growing with increasing number of nodes (how many binary searches, how many corrections were needed, steps were taken)

4 DISCUSSION

- this algorithm has its quirks, but it works

- this is an improvement over the cubic, although the complexity of the algorithm is greatly increased as well

5 FUTURE WORK

- a method for finding regions of monotonicity that is less ad-hoc

- constructing monotone interpolants for higher order approximations

- using improved distribution estimates to improve distribution modeling

6 CONCLUSION

- this paper proposes and tests an algorithm for constructing quintic monotone interpolants

- the results show an improvement over the cubic usage

ACKNOWLEDGMENTS

The NSF is acknowledged for its support of this project.

REFERENCES

Berglund, T., A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist. 2009. "Planning smooth and obstacle-avoiding B-spline paths for autonomous mining vehicles". *IEEE Transactions on Automation Science and Engineering* vol. 7 (1), pp. 167–172.

Brennan, A. 2019, 11. "Measure, Modulation and Metadesign: NC Fabrication in Industrial Design and Architecture". *Journal of Design History*. epz042.

Carlson, R., and F. Fritsch. 1985. "Monotone Piecewise Bicubic Interpolation". *SIAM Journal on Numerical Analysis* vol. 22 (2), pp. 386–400.

Fritsch, F., and R. Carlson. 1980. "Monotone Piecewise Cubic Interpolation". *SIAM Journal on Numerical Analysis* vol. 17 (2), pp. 238–246.

Gregory, J. A. 1985. "Shape preserving spline interpolation". *Brunel University*.

He, X., and P. Shi. 1998. "Monotone B-spline smoothing". *Journal of the American statistical Association* vol. 93 (442), pp. 643–650.

Herman, Daniel Lawrence and Oftedal, Mark J 2015, December. "Techniques and workflows for computer graphics animation system". US Patent 9,216,351.

Hess, W., and J. W. Schmidt. 1994. "Positive quartic, monotone quintic C2-spline interpolation in one and two dimensions". *Journal of Computational and Applied Mathematics* vol. 55 (1), pp. 51–67.

Knott, G. D. 2012. *Interpolating cubic splines*, Volume 18. Springer Science & Business Media.

Piah, A. R. M., and K. Unsworth. 2011. "Improved sufficient conditions for monotonic piecewise rational quartic interpolation". *Sains Malaysiana* vol. 40 (10), pp. 1173–1178.

Quint, A. 2003, July. "Scalable vector graphics". *IEEE MultiMedia* vol. 10 (3), pp. 99–102.

Ramsay, J. O. et al. 1988. "Monotone Regression Splines in Action". *Statistical science* vol. 3 (4), pp. 425–441.

Ulrich, G., and L. Watson. 1994. "Positivity Conditions for Quartic Polynomials". *SIAM Journal on Scientific Computing* vol. 15 (3), pp. 528–544.

Wang, Q., and J. Tan. 2004. "Rational quartic spline involving shape parameters". *Journal of information and Computational Science* vol. 1 (1), pp. 127–130.

Xie, Y., C. B. King, Y. Hong, and Q. Yang. 2018. "Semiparametric models for accelerated destructive degradation test data analysis". *Technometrics* vol. 60 (2), pp. 222–234.

Yao, J., and K. E. Nelson. 2018. "An Unconditionally Monotone C2 Quartic Spline Method with Nonoscillation Derivatives". *Advances in Pure Mathematics* vol. 8 (LLNL-JRNL-742107).

AUTHOR BIOGRAPHIES

THOMAS C. H. LUX is a Ph.D. candidate in Computer Science at Virginia Polytechnic Institute and State University. His research interests include Computational Science, Approximation Theory, Optimization, and Artificial Intelligence. His email address is tchlux@vt.edu.

A AUTHOR CHECKLIST

We strive for a consistent appearance in all papers published in the proceedings. If you used the template and styles within this author's kit, then almost all of the requirements in this checklist will be automatically satisfied, and there is very little to check.

Please **print a hardcopy of your paper**, and go over your printed paper to make sure it adheres to the following requirements. *Thank you!*

1. Abstract
   (a) 150 or fewer words.
   (b) Provide 3-5 keywords (mandatory). This set of keywords will identify your paper in indices and databases.
2. Paper Length
   (a) At least 5, but no more than 12 pages.
   (b) Page size is letter size (8.5" x 11", or 216 mm x 279 mm).
3. All text is in 11-Point Times New Roman except title, header and footer (default in this template).
4. Paper title is in 12-Point Times New Roman **BOLDFACE ALL CAPS** (default in this template).
5. The paper has been spellchecked using U.S. English.
6. Spacing and Margins
   (a) Single spaced.
   (b) Left and right margins are each 1 inch (default in this template).
   (c) Top and bottom margins are according to the template.
   (d) Title starts 1.25 inches from the top of the page (default in this template).
7. Section Headings
   (a) Left justified and set in **BOLDFACE ALL CAPS** (default in this template).
   (b) Numbered, except for the abstract, acknowledgments, references and author biographies.
   (c) Subsection headings are set in **Boldface Headline Style** (default in this template).
8. No footnotes or page numbers.
9. The copyright notice on the first page contains the name of the symposium or a track where you want to submit your paper, and the running head on subsequent pages is the surnames of all authors.
10. Multiple authors are formatted correctly.
11. Equations are centered and any equation numbers are in parentheses and right-justified (default).
12. Figures and Tables
    (a) All text in figures and tables is readable.
    (b) Table captions appear above the table.
    (c) Figure captions appear below the figure.
13. Citations and References
    (a) Citations are by author and year, and are enclosed in parentheses, not brackets (default BIBTEX).
    (b) References are in the hangref style, and are listed alphabetically by the last names(s) of the author(s) (also default BIBTEX setting in this template).
14. Author biographies are one paragraph per author.
15. Hyperlinks
    (a) Be sure that hyperlinks will probably work in the future as well.
    (b) Live hyperlinks are blue. Nonlive hyperlinks are black (default in this template).
16. All fonts must be embedded in the resulting `.pdf`. If you use `pdflatex`, this should already be true–unless you included figures in the `.eps` or `.pdf` format which may introduce additional font dependencies. You can use the `pdffonts` command to see if all the fonts are embedded (the column "emb" should say yes for all rows). You can use GhostScript to force embed the fonts, like so: `gs -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -dEmbedAllFonts=true`

`-sOutputFile=out.pdf -f in.pdf` (in Windows, use `gswin32` or `gswin32c` to invoke GhostScript).

After verifying that your paper meets these requirements, please go to the final submission page at conference website and submit your paper. Be sure to complete the transfer of copyright form and upload the `.pdf` receipt. *Thank you for contributing to the SCS conferences!*

B MOST OBSERVED MISTAKES

The following list comprises **the most common sources of error** that had to be corrected by previous editors. Please make sure to go through the following list and check that your paper is formatted correctly:

1. The paper is less than 5 or more than 12 pages long.
2. Paper title and section titles are in **BOLD ALL CAPS**, subsections are **Bold and Capitalize the First Letter of Important Words**. Please use the templates.
3. Paper is in A4 format, not letter format. Please use the required margins.
4. The copyright notice is incorrect.
5. The running heads are incorrect. Do not forget that the LastNameLastAuthor is preceded by ", and ".
6. The citation format is incorrect. Use BIBTEX for citations and do not change the format used by the template.
7. The biographies are missing. Do not forget the "author biographies" section.
8. Figures or tables are not referenced in the text or have the incorrect caption format.
9. The author section after the title is not formatted correctly, the number of organizations does not define the number of blocks, or the number of blocks does not define the layout.
10. In the heading on the title page, country names are in all capitals.
11. Paragraphs are not indented.
12. Some fonts in the resulting `.pdf` are not embedded.