ROCKY MOUNTAIN JOURNAL OF MATHEMATICS
Vol. , No. , YEAR
https://doi.org/rmj.YEAR..PAGE

PREDICTION INTERVALS FOR INTERPOLANTS

ETHAN WILSON AND ANDREW GARD

ABSTRACT. A classical and well-studied problem in numerical analysis is to find a continuous function that interpolates a discrete set of data points. Most of the theory on this subject assumes that the exact position of each interpolation node is known with certainty, an assumption that is unlikely to hold in scientific practice. This paper analyzes the effects of measurement errors on interpolating functions and provides methods for constructing prediction intervals based upon the distributions of those errors.

1. Introduction

Suppose that we are given a set of observations $(x_0, y_0), \ldots, (x_n, y_n)$ and wish to estimate the response variable y at some new input x. *Interpolation* refers to the construction of a function $f(x)$ that can be used to make such predictions within the range of the known data. It is distinguished from *regression* in that the interpolant is required to map each x_i exactly to its corresponding y_i instead of (or sometimes in addition to) just minimizing a specified cost function. As in the regression setting, a wide variety of interpolation techniques are available depending on the properties desired for the function $f(x)$.

Interpolation is an old problem in the field of numerical analysis. Historians have argued that linear interpolation and perhaps other techniques were used in ancient Babylon [19, 20] and Greece [25] in the third and second centuries B.C.E.. Several centuries later, Claudius Ptolemy (approximately 100-140 C.E.) is known to have relied on interpolation in constructing his astronomical tables [27].

While sophisticated methods were developed in both China and India by at least the seventh century C.E., our modern understanding of polynomial interpolation derives (not surprisingly) from the work of Newton and Lagrange in the late seventeenth century. See [17] for a detailed description of historical developments before and since.

Interpolation has a wide variety of applications. Recent papers have used or adapted interpolation techniques in studies of dissolved oxygen in the ocean [16], ice surface elevation [23], streaming data [5], medical imagery [18], and the bathymetry of Saldhana Bay, South Africa [10], to name just a few.

Of course, all real measurements, whether they be microscopic or astronomical, are inevitably subject to measurement and rounding errors. Such errors are passed on to any interpolating functions based on those measurements, leading to further uncertainty. This paper seeks to quantify that error, considering its propagation through various sorts of interpolants.

Incorporating measures of uncertainty into interpolation models has the potential to substantially improve both the transparency and accuracy of those models. This is illustrated in Figure 1, which shows two polynomial interpolants generated from different sampling runs along the x-axis with a small amount of noise included in the y-direction. The results are quite distinct despite the similarity

2020 *Mathematics Subject Classification.* 65D05, 65D07.
Key words and phrases. Interpolation theory, propagation of error.

of the sample data. Without an understanding in the degree to which variability is inherited or even amplified by the interpolation process, a researcher (who will likely only see only one of these curves) cannot draw conclusions with any degree of reliability. Methods that turn out to be unstable under small errors (like polynomial interpolation, as we will show), should be viewed with skepticism.

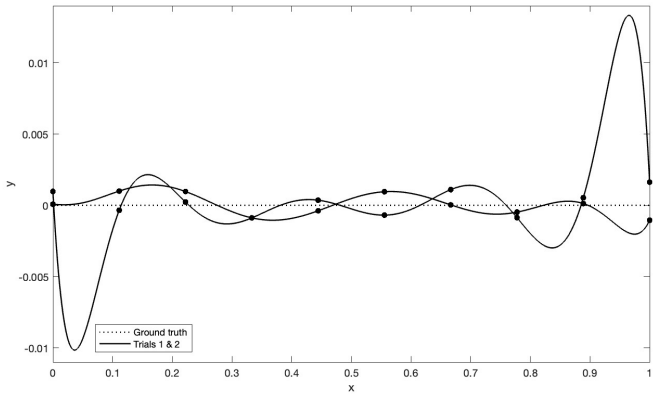

FIGURE 1. Interpolation results based on two different sampling runs

It is of course already standard practice to include measures of variability when reporting single-variable statistics, groupwise comparisons, and regression models, to name just a few. Here we suggest methods for extending these best-practices to interpolation as well.

The effect of measurement uncertainty on interpolants has been considered from various perspectives. Higham [11] performs a stability analysis on polynomial interpolants from the perspective of the condition number, concluding that the barycentric formulation is preferable to the Lagrange form. Hong-ci [13] takes a more purely mathematical real analytical approach to stability. Propagation of error through interpolation has previously been considered in [21] and [22] which address Lagrange and linear interpolation error, respectively, from the perspective of Gaussian processes, each with an aim toward visualization. Finally, Enting, et.al. [6] analyze uncertainty in smoothing splines while examining CH_4 concentrations in ice core samples.

Outside of specific problems like these, error analysis in interpolation theory typically focuses on the situation in which the nodes have been sampled from some fixed "true" function $h(x)$ against which comparisons can be made. For instance, the following error formula for polynomial interpolants is standard [8].

$$h(x) - f_n(x) = \frac{\prod_{k=0}^{n}(x - x_k)}{(n+1)!} h^{(n+1)}(\xi) \tag{1}$$

Here $h(x)$ represents the fixed function, $f_n(x)$ the n^{th}-degree polynomial interpolant, x_k the nodes, and ξ some unknown value in the range of the x_k.

This sort of error analysis is quite different from the problem of measurement uncertainty considered in what follows. In the above formulation, the y-values of the nodes are known with absolute precision and certainty, while in the present work no such assumptions are made, reflecting the more common

experience of practicing scientists. Moreover, we do not posit the existence of any sort of ground-truth function $h(x)$, choosing to focus our attention to the influence of the y-values of the nodes themselves. In this sense, we believe the present work to be novel.

Outside of the interpolation context, propagation of error is a well-developed field and numerous excellent primers are available. See for instance [15] or [24].

In what follows, we explicitly construct prediction intervals for linear, polynomial, and spline interpolants (sections 3, 4, and 5), noting circumstances under which that approach will generalize. In section 6, we apply the Monte Carlo methods of [3] to several additional techniques, including Akima and piecewise-cubic Hermitian polynomials. Section 7 makes numerical comparisons between the various methods, while 8 briefly considers the relationship between prediction intervals for linear regression models and piecewise linear interpolants.

A collection of functions that calculate variances and plot prediction intervals for various interpolating functions is available as a MatLab toolbox (`https://github.com/ethanowilson/Prediction-Intervals-for-Interpolating-Functions.git`).

2. Interpolation as a linear operator

Suppose the observations y_i are randomly distributed with fixed but unknown means and a common variance σ^2, which might for instance represent the finite accuracy of a particular measuring device. For various interpolation methods, we would like to compute the variance of predicted y-values as a function of x and σ^2. While the errors may be normally distributed in many applications, we do not assume that here.

In this paper, we consider several common interpolation methods of the form

$$f(x) = \sum_{i=0}^{n} g_i(x; x_0, \ldots, x_n) y_i \tag{2}$$

that is, ones that are linear in y_i. This family includes piecewise linear, Lagrange polynomial, and cubic spline interpolants, each of which we discuss in turn. For any interpolant of this form,

$$\mathrm{Var}\left(f(x)\right) = \sigma^2 \sum_{i=0}^{n} g_i^2(x; x_0, \ldots, x_n), \tag{3}$$

where $\mathrm{Var}(y_i) = \sigma^2$.

It is clear that the assumption of common variance can be discarded, should the application require.

3. Piecewise Linear Interpolation

Propagation of uncertainty through a line segment interpolanting two points is briefly considered in [22] prior to their larger discussion of Gaussian processes. In this section, we expand on the ideas presented in that discussion as motivation for what follows.

In piecewise linear interpolation, adjacent nodes are connected with straight line segments. That is, for $x \in [x_{i-1}, x_i], 1 \leq i \leq n$,

$$f_i(x) = \frac{x - x_{i-1}}{x_i - x_{i-1}} y_i + \frac{x - x_i}{x_{i-1} - x_i} y_{i-1}. \tag{4}$$

Piecing these together yields the full interpolant $f(x) = \sum_i f_i(x)\chi_{[x_{i-1},x_i]}(x)$. This is clearly linear in each y_i.

For $x \in [x_{i-1}, x_i]$,

$$\mathrm{Var}(f(x)) = \mathrm{Var}\left(\frac{x-x_{i-1}}{x_i-x_{i-1}}y_i + \frac{x-x_i}{x_{i-1}-x_i}y_{i-1}\right) \tag{5}$$

$$= \left(\frac{x-x_{i-1}}{x_i-x_{i-1}}\right)^2 \mathrm{Var}(y_i) + \left(\frac{x-x_i}{x_{i-1}-x_i}\right)^2 \mathrm{Var}(y_{i-1}) \tag{6}$$

$$= \left[\frac{(x-x_{i-1})^2 + (x-x_i)^2}{(x_{i-1}-x_i)^2}\right]\sigma^2. \tag{7}$$

Given any particular x, a level $(1-\alpha)$ prediction intervals for y is then computed using the formula

$$y = f(x) \pm z_* \sigma \sqrt{\frac{(x-x_{i-1})^2 + (x-x_i)^2}{(x_{i-1}-x_i)^2}} \tag{8}$$

where $z_* = \Phi^{-1}\left(1 - \frac{1}{2}\alpha\right)$.

Figure 2 shows a 95% prediction band for eight uniformly distributed nodes on the interval $[0,1]$. Here and elsewhere, all the y_i have been taken to be zero to emphasize the structure of the prediction intervals. The widths of the intervals depend only on x and $x_0, \ldots, x_n$, not on any y_i, so no generality is lost with this assumption.

The structure of the prediction band in this example is entirely typical of piecewise linear interpolation, in which variance is a quadratic function of x that depends only on the immediately adjacent nodes. In fact, the variance computed in line 7,

$$\mathrm{Var}(f(x)) = \frac{\sigma^2}{(x_i-x_{i-1})^2}\left[(x-x_{i-1})^2 + (x-x_i)^2\right], \tag{9}$$

always has its minimum at $x = \frac{1}{2}(x_{i+1}+x_i)$, halfway between the nearest nodes to x.

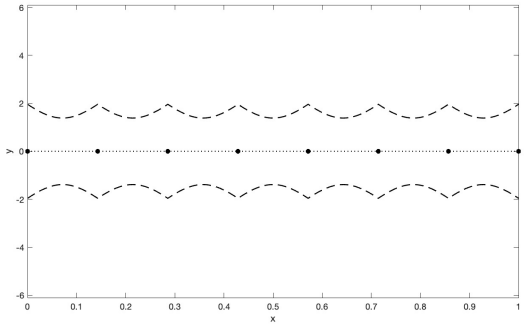

FIGURE 2. 95% prediction interval for a piecewise linear interpolant

4. Lagrange polynomial interpolation

Piecewise linear interpolants are intuitive and simple to compute, but they suffer from a glaring drawback: except for degenerate cases, they fail to be differentiable. This is problematic in any application where a smoothly-varying phenomenon is being modeled. Interpolation using polynomials is the most direct remedy to this problem.

Any $n+1$ nodes with $x_i \neq x_j, i \neq j$ can be joined with a unique polynomial function of degree n [2]. Lagrange's formulation,

$$f(x) = \sum_{k=0}^{n} L_k(x) y_k \tag{10}$$

where

$$L_k(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_n)} \tag{11}$$

is the most transparent, if not always the most computationally efficient. Here $L_k(x_i) = \delta_{ki}$, so each L_k is non-zero at exactly one node.

Following the same logic as in the previous section,

$$\mathrm{Var}(f(x)) = \sigma^2 \sum_{k=0}^{n} L_k^2(x) \tag{12}$$

which yields the prediction interval

$$y = f(x) \pm z_* \sigma \sqrt{\sum_{k=0}^{n} L_k^2(x)} \tag{13}$$

Again, the width of this interval is not dependent on the y_i. Now, however, the dependence on x relative to the x_i is more complicated than in the piecewise linear case, as illustrated in Figure 3. Variance is generally greatest near the endpoints of the interval of interpolation. See section 7 for additional discussion of this phenomenon.

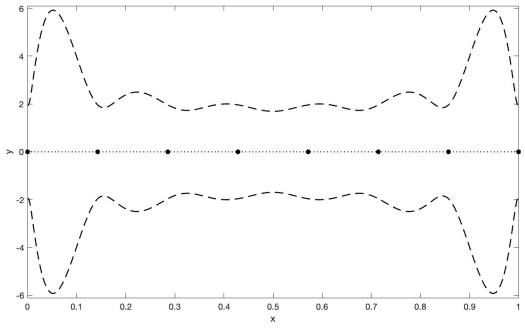

FIGURE 3. 95% prediction interval for a polynomial interpolant

It is also worth noting also that while interpolants themselves can be computationally intensive (Lagrange interpolants are $O(n^2)$ in general [14]), the prediction bands presented in sections 3 and 4 require only a handful of additonal operations. The same can be said of spline interpolants, which we consider next.

5. Cubic spline interpolation

Polynomial interpolation suffers from its own drawbacks. For even moderate numbers of nodes, the functions produced often fluctuate far beyond the range of the observed y_i. Additionally, small changes to those y_i can produce extreme changes to the interpolant near the endpoints of the interval of interpolation, as suggested by Figure 3. Cubic spline interpolation remedies this by fitting low-degree polynomials, which are inherently more stable, to adjacent nodes, then connecting them in a $\mathscr{C}^1$-fashion.

Here we treat only *natural splines,* in which the interpolant is required to have zero second derivative at its endpoints. The calculations for splines with other boundary conditions differ only in computational details. We instead treat these numerically; see section 6.

Given $n+1$ nodes (x_k, y_k), where $k = 0, 1, \ldots, n$, we may write

$$S_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3 \tag{14}$$

for the k^{th} cubic spline polynomial connecting the nodes x_{k-1} and x_k, where now $k = 1, \ldots, n$. Note immediately that $a_k = y_k$ for each polynomial $S_k(x)$.

Our goal is to find $\mathrm{Var}\,(S_k(x))$ for any $x \in [x_{k-1}, x_k]$.

Since

$$\begin{aligned}\mathrm{Var}\,(S_k(x)) = {} & \mathrm{Var}(a_k) + (x - x_k)^2\mathrm{Var}(b_k) + (x - x_k)^4\mathrm{Var}(c_k) + \\ & (x - x_k)^6\mathrm{Var}(d_k) + 2(x - x_k)\mathrm{Cov}(a_k, b_k) + 2(x - x_k)^2\mathrm{Cov}(a_k, c_k) + \\ & 2(x - x_k)^3\mathrm{Cov}(a_k, d_k) + 2(x - x_k)^3\mathrm{Cov}(b_k, c_k) + \\ & 2(x - x_k)^4\mathrm{Cov}(b_k, d_k) + 2(x - x_k)^5\mathrm{Cov}(c_k, d_k),\end{aligned} \tag{15}$$

we must compute the covariance and cross-covariance matrices of the vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$, and $\mathbf{d}$ then extract the diagonal entries.

Set $h_k = x_{k+1} - x_k$ for $k = 1, \ldots, n$. Determining the spline coefficients requires solving the linear system $A\mathbf{c} = \mathbf{v}$, where A is the tridiagonal matrix

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2(h_{n-1} + h_n) & h_n \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \tag{16}$$

and $\mathbf{v}$ is given by

$$\mathbf{v} = \begin{pmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix} \tag{17}$$

See [4] for this, as well as for the formulations of $\mathbf{b}$ and $\mathbf{d}$ used below.

The vector $\mathbf{v}$ can be conveniently written as $M_v\mathbf{a}$, where

$$M_v = \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{3}{h_0} & -\frac{3}{h_0} - \frac{3}{h_1} & \frac{3}{h_1} & 0 & \cdots & 0 & 0 \\ 0 & \frac{3}{h_1} & -\frac{3}{h_1} - \frac{3}{h_2} & \frac{3}{h_2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\frac{3}{h_{n-1}} - \frac{3}{h_n} & \frac{3}{h_n} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{18}$$

Hence $A\mathbf{c} = M_v\mathbf{a}$. This matrix A is diagonal dominant and hence is invertible, allowing us to write

$$\mathbf{c} = A^{-1}M_v\mathbf{a} \tag{19}$$

Next, the condition $d_k = \frac{3}{h_k}(c_{k+1} - c_k)$ can be written $\mathbf{d} = M_d\mathbf{c}$, where

$$M_d = \begin{pmatrix} -\frac{1}{3h_0} & \frac{1}{3h_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{1}{3h_1} & \frac{1}{3h_1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\frac{1}{3h_{n-1}} & \frac{1}{3h_n} \end{pmatrix}, \tag{20}$$

yielding the relation

$$\mathbf{d} = M_dA^{-1}M_v\mathbf{a}. \tag{21}$$

Finally, the relation $b_k = \frac{1}{h_k}(a_{k+1} - a_k) - \frac{h_k}{3}(2c_k + c_{k+1})$ implies that

$$\mathbf{b} = 3M_d\mathbf{a} - M_b\mathbf{c} \tag{22}$$

where

$$M_b = \begin{pmatrix} \frac{2}{3}h_0 & \frac{1}{3}h_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{2}{3}h_1 & \frac{1}{3}h_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{2}{3}h_{n-1} & \frac{1}{3}h_{n-1} \end{pmatrix}. \tag{23}$$

Thus overall

$$\mathbf{b} = \left(3M_d - M_b A^{-1} M_v\right)\mathbf{a} \tag{24}$$

$$\mathbf{c} = \left(A^{-1} M_v\right)\mathbf{a} \tag{25}$$

$$\mathbf{d} = \left(M_d A^{-1} M_v\right)\mathbf{a} \tag{26}$$

Since $\mathbf{a}$ is just a vector of y-values at the interpolation nodes, and errors at the nodes are independent with variance σ^2, the covariance matrix of $\mathbf{a}$ is $K_{\mathbf{aa}} = \sigma^2 I_{n+1}$.

Cross-correlation matrices can now be computed directly. For instance,

$$Cov(\mathbf{b},\mathbf{c}) = Cov\left(\left(3M_d - M_b A^{-1} M_v\right)\mathbf{a}, \left(A^{-1} M_v\right)\mathbf{a}\right) \tag{27}$$

$$= \left(3M_d - M_b A^{-1} M_v\right) Cov(\mathbf{a},\mathbf{a}) \left(A^{-1} M_v\right)^T \tag{28}$$

$$= \left(3M_d - M_b A^{-1} M_v\right) \left(A^{-1} M_v\right)^T \sigma^2 \tag{29}$$

Only the diagonal entries are relevant in the computation of $\mathrm{Var}(S_k(x))$.

The general behavior of a spline interpolant prediction interval is shown in Figure 4. Such intervals appear to be considerably more stable than those of polynomial interpolants, in the sense that uncertainty at the nodes has less tendency to propagate, particularly toward the ends of the interval.

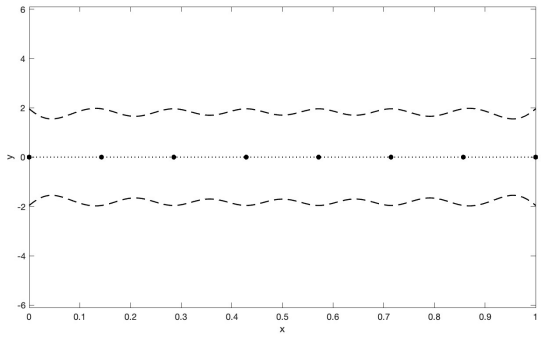

FIGURE 4. 95% prediction interval for a natural cubic spline interpolant

6. Other interpolants

Explicitly calculating pointwise variance as a function of x becomes difficult for more sophisticated interpolation methods, as the previous section illustrates. For interpolation methods that are non-linear in y_i, it may even be impossible to do so exactly. Thus it is desirable to have a numerical alternative.

At least two approaches are possible using Monte Carlo techniques. First, drawing inspiration from the regression setting, we might use repeated simulation to calculate confidence intervals for the coefficients of whatever specific interpolation method is being used. In the case of polynomial interpolation, for instance, this would mean obtaining confidence intervals for each of the n coefficients of the terms of the polynomial interpolant. This approach suffers from a lack of universality, in that

its output will look fundamentally different for each new interpolation method, as well as a lack of interpretability. In the polynomial case, for instance, it is not immediately obvious how a confidence interval for the coefficient of x^3 should be interpreted.

Inspired by [3], we take a more universal approach, generating a large number of interpolants by established methods and setting pointwise prediction intervals using the percentile method. Specifically, for a chosen interpolation method, nodes $(x_0, y_0), \ldots (x_n, y_n)$, and variance σ^2, generate 10,000 interpolants using randomly-generated y-values $\tilde{y}_i \sim N(y_i, \sigma^2)$. At every x in the range of $x_0, \ldots, x_n$, this yields 10,000 predictions for y. A level $(1-\alpha)$ prediction interval is then determined by order statistics $\left(\frac{1}{2}\alpha\right) \cdot (10,000)$ and $\left(1-\frac{1}{2}\alpha\right) \cdot (10,000) + 1$. For instance, if $\alpha = .05$, then the cutoffs are the 250^{th} and $9,751^{st}$ lowest observed values. This is the approached used in our MatLab toolbox, which supports nearest neighbor, modified Akima [1], and piecewise cubic Hermite interpolating polynomial (PCHIP) [9] interpolants, among others.

The actual confidence level of such Monte Carlo prediction intervals is unlikely to be exactly $1-\alpha$. The standard deviation of the confidence level is 0.30% for $\alpha = 0.10$, 0.22% for $\alpha = 0.05$ and 0.10% for $\alpha = 0.01$. See [3] for the computational details.

7. Numerical Experiments

How do the widths of the prediction bands developed in section 3 - 6 change as the number of nodes increases? How do the various methods compare to one another for different distributions of nodes? In this section, we briefly consider each of these questions from an experimental standpoint. It is our hope that with additional work, the observations that follow might be made explicit, generalized, and proven.

As we are only interested in the widths of the prediction bands and not their centers, which are well-established, we will continue to take the y-values at all nodes to be zero. This both simplifies the ensuing plots and makes the relevant phenomena more easily identifiable.

Experimentally, polynomial interpolants show the greatest sensitivity to the number of nodes used, with variance near the endpoints skyrocketing for larger values of n. Figure 5 illustrates this with five, six, and seven uniformly-distributed nodes, respectively. These numbers were chosen for graphical clarity; the phenomenon rapidly becomes more extreme as n continues to increase. This fluctuation near the endpoints is reminiscent of the well-known *Runge phenomenon* in which polynomial interpolants become worse and worse approximators for certain functions as the number of sample points increases (see [7]).

On the other hand, the widths of prediction bands for both piecewise linear and spline interpolants are quite stable as the number of nodes increase. For spline interpolants, variance between nodes is generally (but not always) less than the variance at the nodes. For piecewise linear interpolants, this relationship is strict, as noted in section 3. These phenomena are illustrated in Figure 6, which shows prediction intervals for the two methods with five, six, and seven nodes.

An overlay illustrating the differing widths of the prediction intervals for piecewise linear, polynomial, and spline interpolants is given in Figure 7. Piecewise linear interpolants would appear to be the most stable under sampling uncertainty in the sense that their variances are generally lower than other

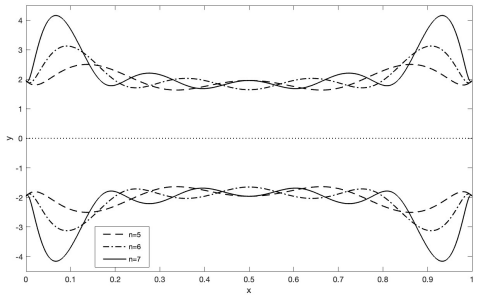

FIGURE 5. 95% prediction intervals for polynomial interpolants with uniform nodes

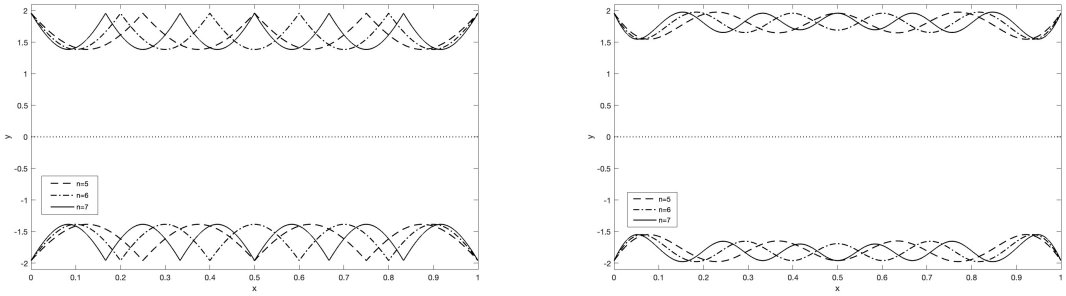

FIGURE 6. 95% prediction intervals for piecewise linear and spline interpolants

methods. Splines are a close second, while polynomial interpolants trailing badly behind, especially toward the endpoints.

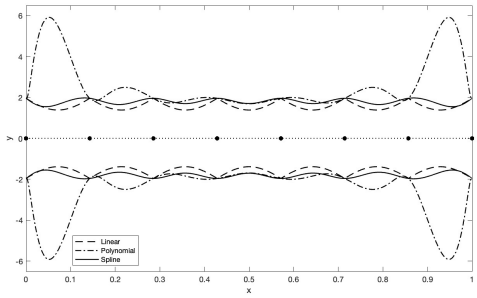

FIGURE 7. 95% prediction bands for various interpolants, regular nodes

We note similar phenomena when the distribution of nodes is not uniform. Figure 8 shows comparisons between the three methods with irregular distributions of five nodes on the unit interval, first clustered to one side and then toward the middle. As in the uniform case, piecewise linear and spline

interpolants remain fairly stable, in the sense that they are less sensitive to perturbations of the nodes, while polynomial interpolants do not.

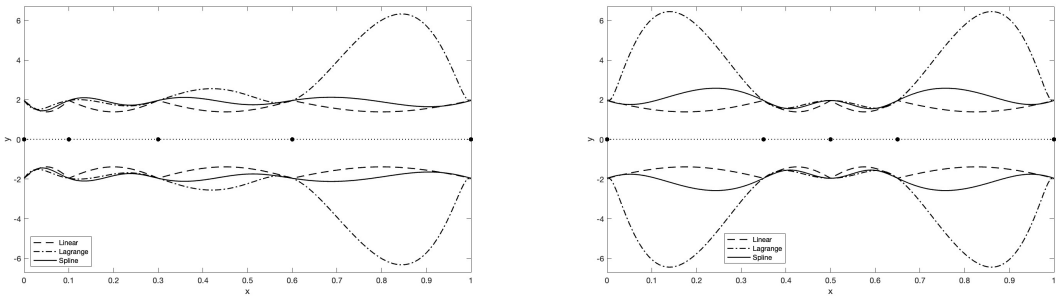

FIGURE 8. 95% prediction bands for various interpolants, irregular nodes

Not surprisingly, the poor reliability of polynomial interpolants is mitigated when nodes can be selected according to the Chebyshev distribution [26], which is known to be optimal with respect to the error formula given in equation 1. This is illustrated in Figure 9.

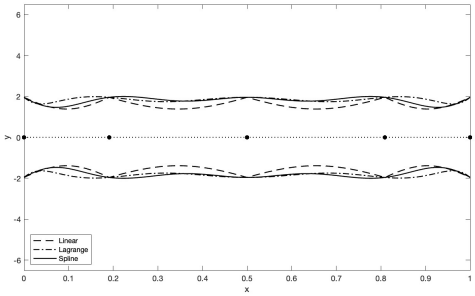

FIGURE 9. 95% prediction bands for various interpolants, Chebyshev nodes

In general, however, it would appear that when observation error may be present in the data, polynomial interpolants are to be avoided in favor or other methods.

Finally, Figure 10 illustrates the variability of two other interpolation methods, modified Akima and PCHIP, under uncertainty at the nodes. On the left, eight nodes are uniformly dispersed. On the right, they have been skewed to one side. In each case, the degree of uncertainty observed is similar to that of a cubic spline interpolant.

8. A Remark on Linear Interpolants and Regression Lines

Let us briefly consider the overlap case between regression and piecewise linear interpolation in which all the sample points $(x_0, y_0), \ldots (x_n, y_n)$ lie on a straight line, comparing the prediction intervals produced by each technique.

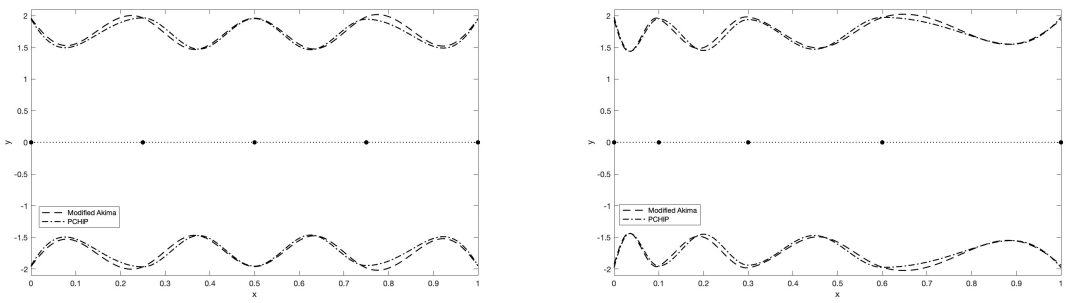

FIGURE 10. 95% prediction bands for two numerical methods

In section 3, we saw (equation 9) that for piecewise linear interpolation,

$$\mathrm{Var}(f(x)) = \frac{\sigma^2}{(x_i - x_{i-1})^2}\left[(x - x_{i-1})^2 + (x - x_i)^2\right], \tag{30}$$

a quantity that depends only on adjacent nodes and which assumes its maximum value σ^2 at those nodes. Here the observed values at the nodes are taken as uncertain while the values predicted by the interpolant are viewed deterministically.

In linear regression, on the other hand, observations at every x-value are assumed to be drawn from a distribution with variance σ^2, regardless of whether the particular x-value exists in the data [12]. Under such circumstances, the width of a prediction interval for a new observation is given by

$$\mathrm{Var}(y) = \sigma^2\left[1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum(x_i - \bar{x})^2}\right]. \tag{31}$$

This expression is strictly greater than σ^2, implying that regression prediction bands are always necessarily wider than interpolation prediction bands. Given that future observations are assumed to have a random component in the regression setting but not the interpolation setting, this result should not be surprising.

Figure 11 illustrates this phenomenon for $n = 8$ uniformly-distributed nodes.

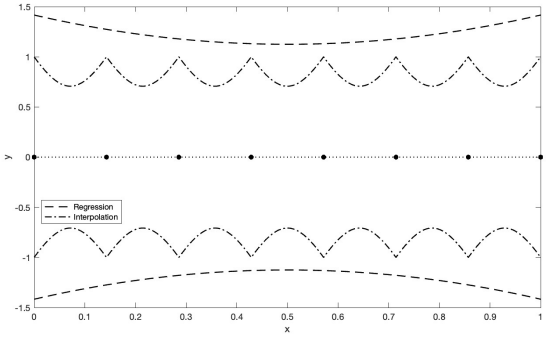

FIGURE 11. Comparison of prediction intervals for regression and interpolation

9. Discussion

The present work provides a roadmap for deepening our understanding of existing interpolation techniques. It includes a full analysis of three historically-important methods and a numerical technique for working with more modern models. Still, many important questions remain unaddressed.

First, a more careful consideration of the role of node distribution in propagation of error is called for. Just how sensitive are the prediction intervals generated by the various techniques to the type and severity of skew? How would the intervals respond to other distributions of nodes? Are certain techniques more universal that others? In Section 7, we briefly considered one skewed distribution of nodes and several symmetric ones, but our choices were arbitrary and the investigation cursory, intended only to suggest further avenues for exploration. For instance, is there a relationship between the optimizing procedure that defines Chebyshev nodes and the narrowed prediction bands that are observed for polynomial interpolants?

A specific analysis of error propagation is needed for each of the wide variety of interpolation methods in use today. While in some practical circumstances, a numerical or visual understanding of the uncertainty inherent in interpolation models might be sufficient, in others, a more theoretical investigation would be useful. In every case, we stand to gain insight into both the interpolants themselves and their sensitivity to observational errors. For instance, while in the present study the distribution of errors (normal or otherwise) played no role in the shape of the resulting prediction intervals, there is no reason to think that fact would hold universally for all interpolation methods.

More broadly speaking, we have few answers to questions about *why* any of the prediction bands fluctuate as they do. For instance, why are these bands so frequently more narrow between nodes than at the observed points? That is, why is there so often more uncertainty at the nodes, where we would seem to have the most information? Only in the piecewise linear case do we have a clear algebraic representation of that phenomenon.

Finally, extension of the present results to grids of two or more dimensions must be considered. Interpolation techniques for higher dimensions rapidly become highly specialized and are often domain specific. While we have chosen not to attempt to address them here, we believe that the techniques applied in this paper provide a workable template for such investigations.

References

[1] Hiroshi Akima. A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the ACM (JACM)*, 17(4):589–602, 1970.

[2] Kendall E Atkinson. *An introduction to numerical analysis*. John wiley & sons, 2008.

[3] Stephen T Buckland. Monte carlo confidence intervals. *Biometrics*, pages 811–817, 1984.

[4] Richard L Burden and J Douglas Faires. *Numerical analysis*. Cengage Learning, 2011.

[5] Roman Dkebski. Real-time interpolation of streaming data. *Computer Science*, 21(4), 2020.

[6] IG Enting, CM Trudinger, and DM Etheridge. Propagating data uncertainty through smoothing spline fits. *Tellus B: Chemical and Physical Meteorology*, 58(4):305–309, 2006.

[7] James F Epperson. On the runge example. *The American Mathematical Monthly*, 94(4):329–341, 1987.

[8] James F Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2021.

[9] Frederick N Fritsch and Ralph E Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980.

[10] Ivan Henrico. Optimal interpolation method to predict the bathymetry of saldanha bay. *Transactions in GIS*, 25(4):1991–2009, 2021.

[11] Nicholas J Higham. The numerical stability of barycentric lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, 2004.

[12] Robert V Hogg, Elliot A Tanis, and Dale L Zimmerman. *Probability and statistical inference*, volume 993. Macmillan New York, 1977.

[13] Huang Hong-Ci. On the stability of interpolation. *Journal of Computational Mathematics*, pages 34–44, 1983.

[14] Ellis Horowitz. A fast method for interpolation using preconditioning. *Information Processing Letters*, 1(4):157–163, 1972.

[15] Ifan Hughes and Thomas Hase. *Measurements and their uncertainties: a practical guide to modern error analysis*. OUP Oxford, 2010.

[16] Takamitsu Ito. Optimal interpolation of global dissolved oxygen: 1965–2015. *Geoscience data journal*, 2021.

[17] Erik Meijering. A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, 2002.

[18] Thiago Moraes, Paulo Amorim, Jorge Vicente Da Silva, and Helio Pedrini. Medical image interpolation based on 3d lanczos filtering. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 8(3):294–300, 2020.

[19] Otto Neugebauer. *A history of ancient mathematical astronomy*, volume 1. Springer Science & Business Media, 2012.

[20] Otto Neugebauer. *Astronomical cuneiform texts: Babylonian ephemerides of the seleucid period for the motion of the sun, the moon, and the planets*, volume 5. Springer Science & Business Media, 2013.

[21] P Saunders and DR White. Propagation of uncertainty due to non-linearity in radiation thermometers. *International Journal of Thermophysics*, 28(6):2098–2110, 2007.

[22] Steven Schlegel, Nico Korn, and Gerik Scheuermann. On the interpolation of data with normally distributed uncertainty for visualization. *IEEE transactions on visualization and computer graphics*, 18(12):2305–2314, 2012.

[23] Undine Strößenreuther, Martin Horwath, and Ludwig Schröder. How different analysis and interpolation methods affect the accuracy of ice surface elevation changes inferred from satellite altimetry. *Mathematical Geosciences*, 52(4):499–525, 2020.

[24] John Robert Taylor and William Thompson. *An introduction to error analysis: the study of uncertainties in physical measurements*, volume 2. Springer, 1982.

[25] GJ Toomer. Hipparchus. *Dictionary of Scientific Biography*, 15:207–224, 1978.

[26] Lloyd N Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. SIAM, 2019.

[27] Glen Van Brummelen. Lunar and planetary interpolation tables in ptolemy's almagest. *Journal for the History of Astronomy*, 25(4):297–311, 1994.

NORTH CAROLINA STATE UNIVERSITY, RALEIGH, NC, 27605, USA
Email address: eowilson@ncsu.edu

LAKE FOREST COLLEGE, LAKE FOREST, IL 60045, USA
Email address: agard@lfc.edu