NOAA Technical Memorandum NOS NGS 94

Interpolation from a Grid of Standard Deviations

Dru Smith

National Geodetic Survey

Silver Spring, MD

May 2023

Versions

Date	Changes
May 2023	Original Release

Contents

1 Introduction

Many models of geodetic phenomena, in particular models created by the National Geodetic Survey (NGS), are represented as data on a spatial grid. Examples include geoid undulations, deflections of the vertical, surface gravity, datum transformations and crustal deformation. In some cases, a second grid is also created, containing standard deviations of the data on the first grid. These paired grids are often used in NGS products or services, in conjunction with an interpolation scheme, to provide values and standard deviations of those values, at points which do not lie on the grid. Though these models often have analytical forms, the direct evaluations often involve intense computations that typical users cannot afford. Interpolating from pre-computed grids saves computation time, though it may introduce additional errors through the process of interpolation itself. But these interpolation errors are controllable and predictable most of the time.

The standard procedure is to interpolate from a 2-D data grid to provide a data estimate at a point of interest in the 2-D domain, and similarly interpolate from the standard deviation grid to provide an estimated standard deviation of the interpolated data value. This paper discusses the mathematical difficulties involved in accurately using a grid of standard deviations to compute the standard deviation of an interpolated value from a grid of data. Through out the rest of the paper, all of the grids are 2-D grids.

2 The mathematics of bilinear interpolation

This paper will use *bilinear* interpolation to examine the right and wrong ways to compute the standard deviation of an interpolated data value. However, the conclusions are easily extendable to other interpolation methods, such as biquadratic (Smith, 2020), which can (like bilinear) be mathematically re-arranged as the weighted sum of values from grid nodes in a finite window that surrounds the interpolation point.

Consider a grid of data values, v, where the grid is spaced evenly in degrees (not meters), with the spacing being $\Delta\phi$ in N/S and $\Delta\lambda$ in E/W. Consider also a companion grid with the same boundaries and spacing, full of standard deviations of v, called σ_v. Now assume that one is interested in an interpolated value of v at some location (ϕ, λ) which will be called the "point of interest" or POI. If bilinear interpolation is used, a 2×2 window[1] $(\Delta\phi \times \Delta\lambda)$ of grid nodes which surrounds (ϕ, λ) is identified and then the interpolation equations are used to interpolate to (ϕ, λ). Although the method of bilinear interpolation is well-established (e.g. Press et al, 1992), it is not always presented as a weighted sum of grid nodes. However, that form is the most useful for the purposes of this paper, and takes this form:

$$v = \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij} v_{ij}. \tag{1}$$

[1] Bilinear interpolation uses a 2×2 window, biquadratic a 3×3 window, etc.

In equation 1,

w_{ij}	are the normalized weights for each grid node,
v_{ij}	are the data values on each grid node, and
v	is the interpolated data value at the POI.

The weights, w_{ij}, are fully derived in the appendix. The data values from the grid nodes are from the 2×2 window around the POI.

By way of example, consider the latitude transformation from NADCON 5.0 (Smith and Bilich, 2017) between NAD 27 and NAD 83(1986) within CONUS (though any product with pairs of data/standard deviation grids would suffice). If the POI is at $(\phi, \lambda) = (40.2°, 290.2°)$, and a 2×2 window has been defined around that point, then the values from the data grid, and from the standard deviation grid, as well as the weights (based upon the formula above) are all shown in Figure 1.

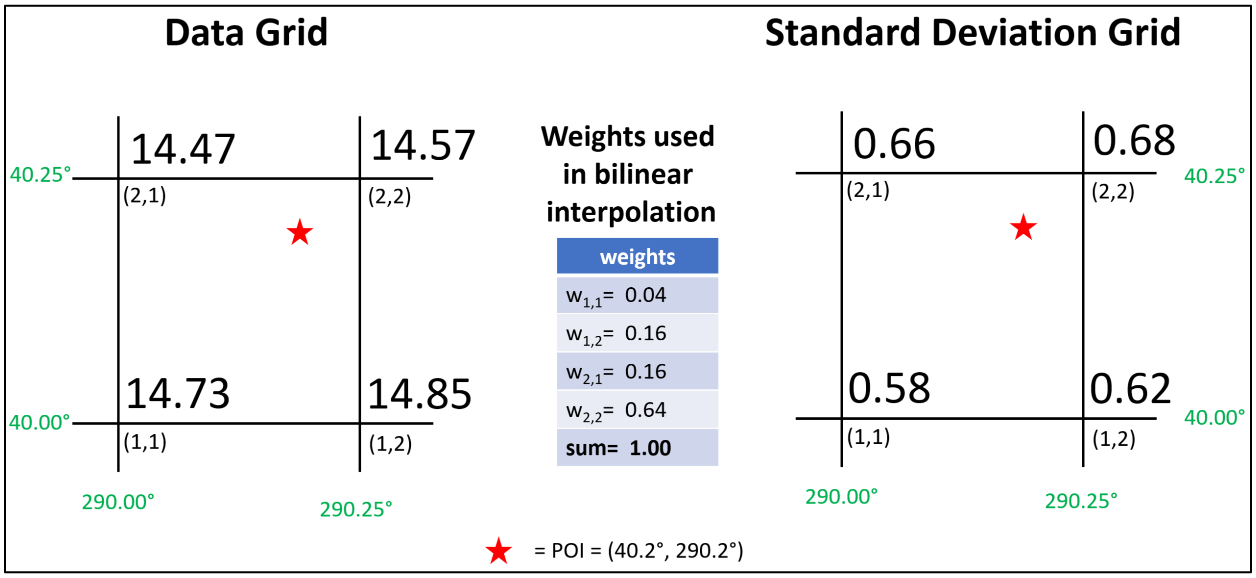

Figure 1: Interpolation windows around a point of interest from a data grid taken from NADCON 5

The bilinearly interpolated *data value* at the POI is easily computed using (1), with the values seen in Figure 1. This computation is:

$$v = (0.04 \times 14.73) + (0.16 \times 14.85) + (0.16 \times 14.47) + (0.64 \times 14.57) = \mathbf{14.61} \qquad (2)$$

Equation 1 is correct for computing an interpolated *data value* (v) from a grid. However, it is *not* correct if it is used to compute the *standard deviation* of the interpolated data value from a grid of standard deviations. That is, if we have a grid of standard deviations provided on the same grid as the original data values, and if σ_v represents standard deviation of v, we claim:

$$\sigma_v \neq \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}\sigma_{v_{ij}}. \tag{3}$$

It should be pointed out that, although mathematically incorrect, many existing and previous NGS products use the right side of (3) to provide users with σ_v.

The correct procedure to compute the standard deviation of an interpolated value is to apply the law of error propagation (Snow, 2021) to (1), including the full use of covariances. *However, the non-equivalence in (3) is true whether covariances exist or not.* To make this point, assume for the moment that no covariances exist, and apply the law of error propagation to (1). This would yield:

$$\begin{aligned}\sigma_v = \sqrt{D\{v\}} = \sqrt{D\left\{\sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}v_{ij}\right\}} = \sqrt{\sum_{i=1}^{2}\sum_{j=1}^{2} D\{w_{ij}v_{ij}\}} = \sqrt{\sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}^2 D\{v_{ij}\}} \\ \neq \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}\sqrt{D\{v_{ij}\}} = \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}\sigma_{v_{ij}}.\end{aligned} \tag{4}$$

Note the "≠" in the middle of (4), a reminder that square roots are not distributive over addition. We therefore conclude that bilinearly interpolating from a grid of standard deviations ($\sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}\sigma_{v_{ij}}$) of some data does not yield the *actual* standard deviation (σ_v) of a bilinearly interpolated value of that data (without regard for what that data is). This same conclusion can be drawn whether the interpolation is bilinear, biquadratic, or any other method that takes the form of a weighted sum of values drawn from the nodes of a grid.

In order to properly compute the standard deviation of the interpolated value, continue to apply the law of error propagation to (1), this time allowing for covariances, as:

$$\begin{aligned}\sigma_v = \sqrt{D\{v\}} = \sqrt{D\left\{\sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}v_{ij}\right\}} = \sqrt{\sum_{i=1}^{2}\sum_{j=1}^{2} D\{w_{ij}v_{ij}\}} \\ = \sqrt{\left(\sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}^2 D\{v_{ij}\}\right) + \left(\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{l=1}^{2} w_{ij}w_{kl}C\{v_{ij},v_{kl}\} \quad \forall\,\{i,j\}\neq\{k,l\}\right)}.\end{aligned} \tag{5}$$

In the case of bilinear interpolation, this means that four variances and six covariances must be known to properly compute the standard deviation, σ_v, of the interpolated data value, v. With a grid of standard deviations, the four variances can easily be computed by squaring the standard deviations. However, if there is no covariance information between the grid nodes, those six covariances will not be available and (5) cannot be properly evaluated. In cases like this, where no covariances are known, it is often the standard procedure to assume they are zero. Section 4 will investigate the magnitude of error which comes with that assumption. However, before doing so, it will be instructive to consider the somewhat philosophical question surrounding the assumption that covariances are zero between standard deviations on grid nodes which all came from the same input data sets and gridding procedure.

3 Does creating grids create correlations and covariances?

The argument is made that if grids of data, and grids of standard deviations of that data, are created from data sources at non-gridded locations, that this process creates distance-dependent correlations in both the data grids and standard deviation grids. If the method of creating the grids is least-squares collocation (Moritz, 1980), then this argument is proven mathematically. However even other methods of gridding should be expected to cause correlations. A cogent defense of this argument is found in Brown (2018).

The method of gridding used to create NADCON 5 grids (both data and standard deviations) was splines-in-tension (Smith and Wessel, 1990), using data at geodetic control points that are obviously not on a grid. This method does not (in fact, *cannot*) produce a covariance function between the gridded standard deviations. However, an argument is made that the standard deviations on the grid *must be* correlated.

The argument goes like this:

Consider a hypothetical situation where the NADCON gridding process somehow created a grid of standard deviations, where the standard deviations on each grid node are *uncorrelated* from one another (thus there are no covariances)[2]. In such a case, the value of standard deviation at any one grid node cannot be predicted by values of standard deviation on other grid nodes. Because of this failure of one grid node value to imply another grid node value, a map of such an uncorrelated grid would show no patterns. In essence, it should look like a "snowy TV screen". However, the choropleth maps of the gridded standard deviations (Smith and Bilich, 2017) absolutely show patterns, primarily driven by both the location and quality of the input data sets. The standard deviation grid for latitude transformations from NAD 27 to NAD 83(1986) in CONUS has been reproduced in Figure 2.

[2] How such a situation might occur is admittedly difficult, if not impossible, to imagine. Nonetheless, the near impossibility of the hypothesis is what lends weight to the argument in the end.

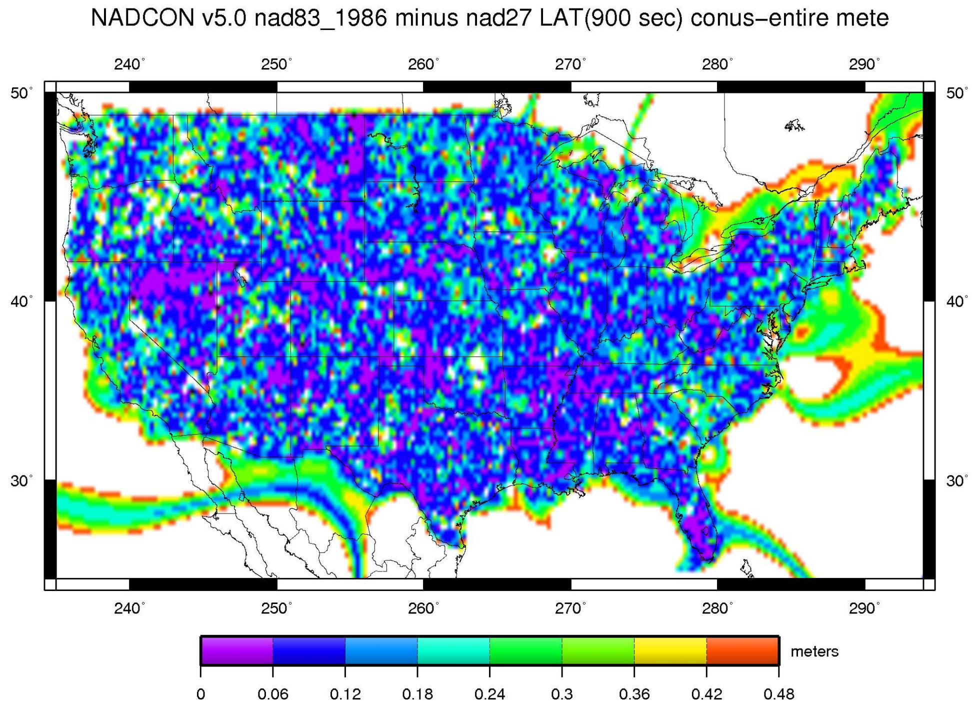

Figure 2: Gridded standard deviations in NADCON 5 for latitude shifts from NAD 27 to NAD 83(1986) in CONUS

Since these patterns exist, the hypothetical case of "no covariances" is rejected, and it is concluded that *some covariances must exist between the standard deviations on grid nodes*.

Having concluded that correlations (and thus covariances) must exist in a grid of standard deviations, it is unfortunately true that in *all cases* of current NGS products and services that the correlation function, and its associated covariance function, are not available.

But can those functions be deduced? Consider the question: Given only a grid of data and a grid of the standard deviations of that data, with no further information, is it possible to derive the correct covariances (or even a simple empirical estimate of them) which should be used in (5)? This question was posed to various geodetic experts (Dennis Milbert, personal communication; Kyle Snow, personal communication; Nicholas Brown, personal communication) and the results, though not conclusive, seem to point more to "no" than "yes". Clearly further investigation into this question is warranted, but it is not pursued in this memorandum.

But the following conundrum still remains: *Non-zero correlations (and thus covariances) must exist for any gridding process which creates a grid of standard deviations, but in the absence of believable correlation (or covariance) information, covariances often are treated as zero.*

This conundrum cannot be resolved without some knowledge of the correlations and/or covariances. Assuming that knowledge remains elusive, the best that can be done is to understand the *impact* of assuming zero covariances, and determine on a case-by-case basis whether the numerical impact of that assumption is acceptable. The next section explores the situation through numerical examples.

4 Some numerical examples

Consider the gridded data and standard deviations in Figure 1, and the desire to compute the true standard deviation of the interpolated value, σ_v, using (5). Missing from (5) are the six covariances representing all combinations of pairs of grid nodes. As mentioned earlier, the correlations (and thus covariances) between any pair of grid nodes may be simple, complicated, or something in between, but in general it is clear that they are *not zero*. Without further information to go on, in order to better understand the impact of setting covariances to zero, we will consider four hypothetical cases: that the grid nodes are (a) strongly correlated (b) moderately correlated (c) weakly correlated or (d) uncorrelated. This last, though argued to be untrue (section 3) is also the most frequently used assumption and will serve as a numerical baseline against the other three cases.

Without attempting to justify numbers with data, Table 1 will reflect the four cases. Correlations are used, rather than covariances, as they are always bounded in the domain +1 to -1, whereas covariances will carry a range that is reflective of the variances themselves.

Table 1: Simulated correlations for four possible cases

	Strong	**Moderate**	**Weak**	**None**
$\boldsymbol{\rho_{11,12}}$	0.99	0.50	0.10	0.00
$\boldsymbol{\rho_{11,21}}$	0.99	0.50	0.10	0.00
$\boldsymbol{\rho_{11,22}}$	0.95	0.40	0.05	0.00
$\boldsymbol{\rho_{12,21}}$	0.95	0.40	0.05	0.00
$\boldsymbol{\rho_{12,22}}$	0.99	0.50	0.10	0.00
$\boldsymbol{\rho_{21,22}}$	0.99	0.50	0.10	0.00

Note that in all cases, the correlations along the shorter window edges were assumed higher than correlations across the longer diagonals of the window, reflecting an assumed falling-off of the correlation function with longer distances.

From these values, and the variances, covariances can be computed from this equation:

$$\sigma_{v_{ij},v_{kl}} = \rho_{ij,kl}\sigma_{ij}\sigma_{kl} \tag{6}$$

Inserting the correlations seen in Table 1, and the standard deviations at grid nodes seen in Figure 1, into (6), the covariances for the four cases were computed and are shown in Table 2.

Table 2: Simulated covariances for four cases

	Strong	**Moderate**	**Weak**	**None**
$\boldsymbol{\sigma_{v_{11},v_{12}}}$	0.356004	0.17980	0.03596	0.00
$\boldsymbol{\sigma_{v_{11},v_{21}}}$	0.378972	0.19140	0.03828	0.00
$\boldsymbol{\sigma_{v_{11},v_{22}}}$	0.374680	0.15776	0.01972	0.00
$\boldsymbol{\sigma_{v_{12},v_{21}}}$	0.388740	0.16368	0.02046	0.00
$\boldsymbol{\sigma_{v_{12},v_{22}}}$	0.417384	0.21080	0.04216	0.00
$\boldsymbol{\sigma_{v_{21},v_{22}}}$	0.444312	0.22440	0.04488	0.00

These covariances can be inserted into (5) to yield σ_v. However, note in (5) that the weighted sum of the *variances* will be identical, no matter what the covariances are. For conciseness, call this portion "q". That is, using the weights and standard deviations (squared to make variances) of the four grid nodes, as seen in Figure 1, we can compute the variance-specific portion of (5), q:

$$\begin{aligned} q &= \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}^2 D\{v_{ij}\} \\ &= (0.0016 \times 0.3364) + (0.0256 \times 0.3844) + (0.0256 \times 0.4356) \\ &\quad + (0.4096 \times 0.4624) = \mathbf{0.210929}. \end{aligned} \tag{7}$$

We now write the *covariance* portion of (5), which will be called r, as:

$$\begin{aligned} r &= \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}\sum_{l=1}^{2} w_{ij}w_{kl}C\{v_{ij}, v_{kl}\} \quad \forall \{i,j\} \neq \{k,l\} \\ &= w_{11}w_{12}C\{v_{11}, v_{12}\} + w_{11}w_{21}C\{v_{11}, v_{21}\} + w_{11}w_{22}C\{v_{11}, v_{22}\} \\ &\quad + w_{12}w_{11}C\{v_{12}, v_{11}\} + w_{12}w_{21}C\{v_{12}, v_{21}\} + w_{12}w_{22}C\{v_{12}, v_{22}\} \\ &\quad + w_{21}w_{11}C\{v_{21}, v_{11}\} + w_{21}w_{12}C\{v_{21}, v_{12}\} + w_{21}w_{22}C\{v_{21}, v_{22}\} \\ &\quad + w_{22}w_{11}C\{v_{22}, v_{11}\} + w_{22}w_{12}C\{v_{22}, v_{12}\} + w_{22}w_{21}C\{v_{22}, v_{22}\} \\ &= \mathbf{2[w_{11}w_{12}C\{v_{11}, v_{12}\} + w_{11}w_{21}C\{v_{11}, v_{21}\} + w_{11}w_{22}C\{v_{11}, v_{22}\}} \\ &\quad \mathbf{+ w_{12}w_{21}C\{v_{12}, v_{21}\} + w_{12}w_{22}C\{v_{12}, v_{22}\} + w_{21}w_{22}C\{v_{21}, v_{22}\}]}. \end{aligned} \tag{8}$$

The value of r depends on which case is being considered. Using the covariances from the above table, each value of q was computed, added to r and then a square root taken to compute the standard deviation. The results are in Table 3.

Table 3: Contributions of variances, covariances, their sum, and final standard deviation of interpolated value v

	Strong	**Moderate**	**Weak**	**None**
q	0.210929	0.210929	0.210929	0.210929
r	0.224970	0.110338	0.0208333	0.000000
$q + r$	0.435899	0.321267	0.2317623	0.210929
σ_v	**0.660**	**0.567**	**0.481**	**0.459**

Table 3 contains formally computed standard deviations of the interpolated data value, using (5), based on a-priori weights, variances and four different (assumed known) sets of correlations.

There is, of course, another method for computing (or, more accurately, "approximating") the standard deviation of an interpolated data value, and that is to simply use bilinear interpolation from the grid of standard deviations, using (1). That method is, as mentioned earlier, encoded in many NGS products and services. To use that method, we apply the weights and gridded standard deviations seen in Figure 1 to (1) and get the approximate standard deviation seen in (9).

$$\sigma_v = (0.04 \times 0.58) + (0.16 \times 0.62) + (0.16 \times 0.66) + (0.64 \times 0.68) = \mathbf{0.663}. \tag{9}$$

From (9) and Table 3, certain conclusions can be drawn.

The first is that the value of σ_v in (9), is very close to that in the "strongly correlated" case. In fact, if the correlations between grid nodes were actually set exactly to 1.0, then the value of σ_v obtained through (5) would be exactly 0.663, perfectly matching that from (9).

This leads to a second conclusion: computing σ_v by bilinearly interpolating from the grid of standard deviations, though called a "mathematical difficulty" earlier in the paper, could more accurately be called "mathematically correct *if and only if* all of the standard deviations in each grid node were perfectly correlated with each other grid node." As it seems unlikely that perfect correlation will always exist between all grid nodes, and yet it also seems likely that the correlation between grid nodes may be quite high, it may be concluded that computing σ_v by using bilinear interpolation from a grid of standard deviations is *reasonable, though slightly too pessimistic*.

The third conclusion is this: if the correlations between standard deviations on grid nodes are treated as exactly zero (a common assumption in the absence of true correlation information), then the computed σ_v value, using the rigorous method in (5), will yield the smallest σ_v values in all of these examples. However, as argued in section 3, it is impossible for the standard deviations to be completely uncorrelated in most gridding methods. In fact, it is re-iterated that the correlation is likely to be quite high. Therefore, it is concluded that treating the correlations as zero yields a value of σ_v that is significantly too optimistic.

Therefore, in the absence of true correlation (or covariance) functions, and based solely on the reasoning above, it is concluded that:

> *the application of bilinear interpolation to a standard deviation grid, while slightly pessimistic, yields a value of σ_v closer to truth than formally computing it under the assumption of no covariances.*

5 Biquadratic and other interpolations

The previous conclusions can all be derived from other interpolation methods, provided they can be written as some weighted sum of values on grids. For example, biquadratic interpolation (Smith, 2020), which uses a 3 × 3 grid around a POI, can be rearranged in the following way:

$$v = \sum_{i=1}^{3}\sum_{j=1}^{3} w_{ij} v_{ij}. \tag{10}$$

The equations for computing the weights in (10) are more complicated than those for bilinear and can be found in the appendix. As before, the proper way to get σ_v for an interpolated v value that comes from (10), is to apply error propagation to (10), resulting in 9 variances and 36 covariances, and using a known covariance function. But, if that function is unavailable, then biquadratically interpolating off of a grid of standard deviations will, like bilinear, yield a value that is slightly pessimistic but also closer to the truth than using formal error propagation and setting the covariances to zero.

6 Application to differences

Briefly, the idea of differences of interpolated values is addressed. This is done only to introduce the topic, for it is far more complicated than that of a single point, and requires its own study and report.

Consider again some grid of data and its grid of standard deviations, and consider two points of interest, A and B, each within the bounds of the grids, but each separated by a distance far enough that their interpolation windows are different. If the interpolated values at A and B are v_A and v_B, and someone were interested in not only their difference, but also the standard deviation of that difference, then some significant difficulties arise.

For now, just to avoid the complicating factors that come with interpolation, let us assume that A and B happen to fall *exactly* on grid nodes. Then, the standard deviation of the difference between v_A and v_B is:

$$\sigma_{v_B - v_A} = \sqrt{D\{v_B - v_A\}} = \sqrt{D\{v_A\} + D\{v_B\} - 2C\{v_A, v_B\}} \tag{11}$$

where

$$C\{v_A, v_B\} = \rho_{v_A, v_B}\sqrt{D\{v_A\}}\sqrt{D\{v_B\}} \tag{12}$$

In the previous section, where the correlation needed to be considered only over *very short distances* (just between grid nodes in an interpolation window), it was easy enough to rationalize that the correlation coefficient would be "quite high". However, in this case, no such simple assumption can be taken. It cannot be said, in general, whether the correlation in the above equation is high, medium, low, zero, nor whether it is positive or negative.

In nearly every case where a correlation is unknown, it is set to zero, along with the covariance. And as can be seen in (11), setting the covariance to zero means that a standard deviation of a difference of values will be too large (if the covariance is positive) or too small (if the covariance is negative).

Nothing further will be said in this report about this issue, except this: when creating gridded models, it behooves the modeler to document the covariance and correlation functions and provide them as part of the gridded data and gridded standard deviations. Barring that, either some method of empirically computing the proper covariances from the two grids must be found, or the assumption of a zero covariance must be used and its consequences accepted.

7 Conclusions

This memorandum has explored the problem of correctly determining the standard deviation of a value that has been interpolated off of a grid, assuming that a grid of data and a grid of standard deviations of that data exist. It was stated that the mathematically correct way to do so was to apply error propagation to the interpolation equation itself. However, this required knowledge of the covariances between standard deviations at grid nodes, something generally unavailable.

It was deduced that such a covariance should not be zero, though it is often a common practice to assume covariances *are* zero when no other information about them exists. It was further postulated that, at least in the short distances of grid nodes, a correlation should not only exist but be close to 1.0. However, no method could be found to adequately compute the true correlation function based solely upon the two grids.

Further, it was demonstrated that the bilinear interpolation equation itself, if applied to gridded standard deviations, inherently carries with it the assumption that the correlations between grid nodes must be exactly 1.0.

Therefore, the conclusion of this report is that, in the absence of true information about covariances and correlations, interpolating from a grid of standard deviations, while slightly pessimistic, is both easy to do and provides an estimate of the true standard deviation of the interpolated data value that is closer to the truth than using formal error propagation and setting the covariances to zero.

8 Bibliography

Brown, N, et al., 2018: *AUSGeoid2020 combined gravimetric–geometric model: location-specific uncertainties and baseline-length-dependent error decorrelation*, Journal of Geodesy, v. 92, pp. 1457-1465.

Moritz, H., 1980: *Advanced Physical Geodesy*, Second Edition, Herbert Wichman Verlag GmbH, Karlsruhe.

Press, W. H, S.A. Teukolsky, W.T. Vetterling and B. P. Flannery, 1992: *Numerical Recipes in FORTRAN*, Second Edition, Cambridge University Press.

Snow, K., 2021. Adjustment Computations: OSU Class Notes [Based on former courses taught by Burkhard Schaffrin at The Ohio State University]. https://earthsciences.osu.edu/research/geodetic-science/resources

Smith, D. and A. Bilich, 2017: *NADCON 5.0: Geometric Transformation Tool for Points in the National Spatial Reference System*, NOAA Technical Report NOS NGS 63, https://geodesy.noaa.gov/library/pdfs/NOAA_TR_NOS_NGS_0063.pdf

Smith, D., 2020: *Biquadratic Interpolation*, NOAA Technical Memorandum NOS NGS 84. https://geodesy.noaa.gov/library/pdfs/NOAA_TM_NOS_NGS_0084.pdf

Smith, W. H. F, and P. Wessel, 1990: *Gridding with continuous curvature splines in tension*, Geophysics, 55, 293-305.

9 Appendix: Deriving the weight-based interpolation equations

The equations for interpolating from a grid, when using a window of points that is a sub-set of the entire grid, can generally always be broken down into the weighted sum of the values at each grid node of the window. This appendix will derive those weights for the two most popular interpolation algorithms in NGS products and services, bilinear and biquadratic.

Though general discussion of interpolation from a grid often uses X and Y as the axes, this appendix will use longitude, λ, rather than X and latitude, ϕ, rather than Y. The terms "evenly spaced" will mean that separations between parallels, $\Delta\phi$, will be constant (in degrees) and separations between meridians, $\Delta\lambda$, will be constant (in degrees), and not that they are constant in *distance*. Thus, convergence of the meridians is ignored.

9.1 Bilinear

The bilinear interpolation equations are fairly well established. A complete discussion of them can be found in Press et al. (1992). Because it will be convenient to draw parallels to the next section (biquadratic), this section will approach bilinear in a very specific way. For reference, see Figure 1 from earlier in this report.

The steps for bilinear interpolation can be summed up as:

1) Begin with a grid of evenly spaced data, bounded by ϕ_N, ϕ_S, λ_W, λ_E and with spacings of $\Delta\phi$ and $\Delta\lambda$. Each value at a grid node will be given the name $v_{i,j}$, for each row (i) and column (j) of the node.
2) Choose a point of interpolation, POI, (ϕ, λ) which falls within the grid boundaries.
3) Find a 2×2 window of evenly spaced gridded data around the POI. Give the latitudes, longitudes and values of the grid nodes these variables: (ϕ_1, λ_1) with data value $v_{1,1}$, (ϕ_1, λ_2) with data value $v_{1,2}$, (ϕ_2, λ_1) with data value $v_{2,1}$ and (ϕ_2, λ_2) with data value $v_{2,2}$.
4) Fit an east-west line through the southernmost two points. Find the point on that line corresponding to λ, which we call point A. Evaluate that line at A to get v_A, at (ϕ_1, λ).
5) Fit an east-west line through the northernmost two points. Find the point on that line corresponding to λ, which we call point B. Evaluate that line at B to get v_B, at (ϕ_2, λ).
6) Fit a north-south line through points A and B. Find the point on that line corresponding to ϕ, which is our POI. Evaluate that line at the POI to get v, at (ϕ, λ).

The mathematics of these steps yields the following equations.

Let any general line be written as:

$$y = a + bx. \tag{13}$$

Then for the south line:

$$a_1 = \frac{1}{\Delta\lambda}\left(v_{1,1}[\lambda_1 + \Delta\lambda] - v_{1,2}\lambda_1\right), \tag{14}$$

$$b_1 = \frac{1}{\Delta\lambda}\left(v_{1,2} - v_{1,1}\right), \tag{15}$$

$$v_A = \frac{1}{\Delta\lambda}\left(v_{1,1}[\lambda_1 + \Delta\lambda] - v_{1,2}\lambda_1\right) + \frac{1}{\Delta\lambda}\left(v_{1,2} - v_{1,1}\right)\lambda = v_{1,1} - \frac{(\lambda - \lambda_1)\left(v_{1,1} - v_{1,2}\right)}{\Delta\lambda}. \tag{16}$$

And for the north line:

$$a_2 = \frac{1}{\Delta\lambda}\left(v_{2,1}[\lambda_1 + \Delta\lambda] - v_{2,2}\lambda_1\right), \tag{17}$$

$$b_2 = \frac{1}{\Delta\lambda}\left(v_{2,2} - v_{2,1}\right), \tag{18}$$

$$\begin{aligned} v_B &= \frac{1}{\Delta\lambda}\left(v_{2,1}[\lambda_1 + \Delta\lambda] - v_{2,2}\lambda_1\right) + \frac{1}{\Delta\lambda}\left(v_{2,2} - v_{2,1}\right)\lambda \\ &= v_{2,1} - \frac{(\lambda - \lambda_1)\left(v_{2,1} - v_{2,2}\right)}{\Delta\lambda}. \end{aligned} \tag{19}$$

And the final line through points A and B:

$$a = \frac{1}{\Delta\phi}(v_A[\phi_1 + \Delta\phi] - v_B\phi_1), \tag{20}$$

$$b = \frac{1}{\Delta\phi}(v_B - v_A), \tag{21}$$

$$\begin{aligned} v &= \frac{1}{\Delta\phi}(v_A[\phi_1 + \Delta\phi] - v_B\phi_1) + \frac{1}{\Delta\phi}(v_B - v_A)\phi \\ &= \frac{(\Delta\phi - \phi + \phi_1)(\Delta\lambda - \lambda + \lambda_1)}{\Delta\phi\Delta\lambda}v_{1,1} + \frac{(\Delta\phi - \phi + \phi_1)(\lambda - \lambda_1)}{\Delta\phi\Delta\lambda}v_{1,2} \\ &+ \frac{(\phi - \phi_1)(\Delta\lambda - \lambda + \lambda_1)}{\Delta\phi\Delta\lambda}v_{2,1} + \frac{(\phi - \phi_1)(\lambda - \lambda_1)}{\Delta\phi\Delta\lambda}v_{2,2}. \end{aligned} \tag{22}$$

Thus, the bilinear interpolation weights can be inferred as:

$$w_{1,1} = \frac{(\Delta\phi - \phi + \phi_1)(\Delta\lambda - \lambda + \lambda_1)}{\Delta\phi\Delta\lambda}, \tag{23}$$

$$w_{1,2} = \frac{(\Delta\phi - \phi + \phi_1)(\lambda - \lambda_1)}{\Delta\phi\Delta\lambda}, \tag{24}$$

$$w_{2,1} = \frac{(\phi - \phi_1)(\Delta\lambda - \lambda + \lambda_1)}{\Delta\phi\Delta\lambda}, \tag{25}$$

$$w_{2,2} = \frac{(\phi - \phi_1)(\lambda - \lambda_1)}{\Delta\phi\Delta\lambda}. \tag{26}$$

Therefore, bilinear interpolation may be reduced to:

$$v_{bilinear} = \sum_{i=1}^{2}\sum_{j=1}^{2} w_{ij}v_{ij} \tag{27}$$

using the weights above. Note that the sum of the bilinear weights will always equal 1.

9.2 Biquadratic

The method of biquadratic interpolation is less well documented in the literature than bilinear, though a complete exposition of it can be found in Smith (2020). Missing from that document, though, is the restructuring of the biquadratic interpolation method into the sum of weighted grid nodes. This section will summarize restructuring, much in the same way as in the previous section. However, as the details get a bit tedious, some will be excluded.

The basic

The steps for biquadratic interpolation can be summed up as:

1) Begin with a grid of evenly spaced data, bounded by ϕ_N, ϕ_S, λ_W, λ_E and with spacings of $\Delta\phi$ and $\Delta\lambda$. Each value at a grid node will be given the name $v_{i,j}$, for each row (i) and column (j) of the node.
2) Choose a point of interpolation, POI, (ϕ, λ) which falls within the grid boundaries.
3) Find a 3 × 3 window (see Figure 3) of evenly spaced gridded data around the POI. Give the latitudes, longitudes and values of the grid nodes the variables: (ϕ_i, λ_j) with value $v_{i,j}$ for all $\{i, j\} \in \{1,2,3\}$.
4) Fit an east-west parabola through the southernmost three points. Find the point on that parabola corresponding to λ, which we call point A. Evaluate that parabola at A to get v_A, at (ϕ_1, λ).
5) Fit an east-west parabola through the middle three points. Find the point on that parabola corresponding to λ, which we call point B. Evaluate that parabola at B to get v_B, at (ϕ_2, λ).

6) Fit an east-west parabola through the northernmost three points. Find the point on that parabola corresponding to λ, which we call point C. Evaluate that parabola at C to get v_C, at (ϕ_3, λ).
7) Fit a north-south parabola through points A, B and C. Find the point on that parabola corresponding to ϕ, which is our POI. Evaluate that parabola at the POI to get v, at (ϕ, λ).

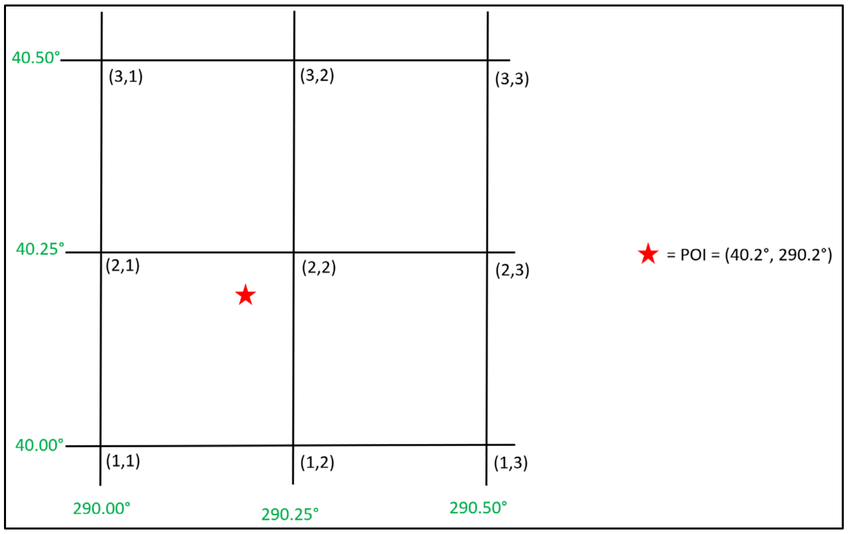

Figure 3: An example of a 3 × 3 window used in biquadratic interpolation.

The mathematics of these steps yields the following equations.

Let any general parabola be written as:

$$y = a + bx + cx^2. \tag{28}$$

For the southernmost parabola:

$$a_1 = \frac{1}{2\Delta\lambda^2}\left[v_{1,1}\left(2\Delta\lambda^2 + 3\lambda_1\Delta\lambda + {\lambda_1}^2\right) + v_{1,2}\left(-4\lambda_1\Delta\lambda - 2{\lambda_1}^2\right) + v_{1,3}\left(\lambda_1\Delta\lambda + {\lambda_1}^2\right)\right], \tag{29}$$

$$b_1 = \frac{1}{2\Delta\lambda^2}\left[v_{1,1}(-3\Delta\lambda - 2\lambda_1) + v_{1,2}(4\Delta\lambda + 4\lambda_1) + v_{1,3}(-\Delta\lambda + -2\lambda_1)\right], \tag{30}$$

$$c_1 = \frac{1}{2\Delta\lambda^2}\left[v_{1,1}(1) + v_{1,2}(-2) + v_{1,3}(1)\right], \tag{31}$$

$$v_A = \frac{1}{2\Delta\lambda^2}\left[v_{1,1}(\Delta\lambda - \lambda + \lambda_1)(2\Delta\lambda - \lambda + \lambda_1) + v_{1,2}(\lambda + \lambda_1)(2\Delta\lambda - \lambda + \lambda_1) - v_{1,3}(\lambda + \lambda_1)(\Delta\lambda - \lambda + \lambda_1)\right]. \tag{32}$$

The process continues, as described above until the value v is evaluated from the final north-south parabola. Simplifying the equation yields the weights seen in (33) through (41).

$$w_{1,1} = \frac{(\Delta\lambda - \lambda_0 + \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\Delta\phi - \phi_0 + \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{4\Delta\lambda^2\Delta\phi^2} \tag{33}$$

$$w_{1,2} = \frac{(\lambda_0 - \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\Delta\phi - \phi_0 + \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{2\Delta\lambda^2\Delta\phi^2} \tag{34}$$

$$w_{1,3} = -\frac{(\lambda_0 - \lambda_1)(\Delta\lambda - \lambda_0 + \lambda_1)(\Delta\phi - \phi_0 + \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{4\Delta\lambda^2\Delta\phi^2} \tag{35}$$

$$w_{2,1} = \frac{(\Delta\lambda - \lambda_0 + \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{2\Delta\lambda^2\Delta\phi^2} \tag{36}$$

$$w_{2,2} = \frac{(\lambda_0 - \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{\Delta\lambda^2\Delta\phi^2} \tag{37}$$

$$w_{2,3} = -\frac{(\lambda_0 - \lambda_1)(\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(2\Delta\phi - \phi_0 + \phi_1)}{2\Delta\lambda^2\Delta\phi^2} \tag{38}$$

$$w_{3,1} = -\frac{(\Delta\lambda - \lambda_0 + \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(\Delta\phi - \phi_0 + \phi_1)}{4\Delta\lambda^2\Delta\phi^2} \tag{39}$$

$$w_{3,2} = -\frac{(\lambda_0 - \lambda_1)(2\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(\Delta\phi - \phi_0 + \phi_1)}{2\Delta\lambda^2\Delta\phi^2} \tag{40}$$

$$w_{3,3} = \frac{(\lambda_0 - \lambda_1)(\Delta\lambda - \lambda_0 + \lambda_1)(\phi_0 - \phi_1)(\Delta\phi - \phi_0 + \phi_1)}{4\Delta\lambda^2\Delta\phi^2} \tag{41}$$

Thus, analogous to bilinear interpolation, biquadratic interpolation may be reduced to a weighted sum of grid values, now seen in (42), and using the nine weights listed above.

$$v_{biquadratic} = \sum_{i=1}^{3}\sum_{j=1}^{3} w_{ij}v_{ij} \tag{42}$$

As with bilinear, the sum of weights in biquadratic interpolation will always equal 1.