The Random-Walk Interpolation Algorithm

J. W. Fowler

10 September, 2004

This memo describes an algorithm for estimating the value of a parameter at an arbitrary epoch given two measurements at different epochs when the mechanism that causes the value to change is completely unknown and therefore cannot be modeled. It is assumed that if the variation *can* be modeled, then it *will* be, and the parameter of interest is a coefficient (or a set of coefficients) of this model which should not change but does anyway. When the need for such an algorithm arises, as it occasionally does when operating recently developed observing equipment with unfamiliar properties, greater uncertainties must be accepted as a natural consequence. For example, when the dark current of a temperature-stabilized infrared imaging array is not expected to vary with time but nevertheless does, either it must be measured every time the array is used, or some way of estimating its value from other measurements must be employed. The former possibility is frequently impractical, leaving some form of interpolation as the best available method for obtaining a value for the epoch needed. This inevitably involves hoping that the existing measurements do not undersample the variation too severely. But there is no solid justification for making that assumption, and so it becomes necessary to provide some reasonable estimate for the uncertainty of the interpolated value. By "reasonable" we mean neither badly overestimated nor badly underestimated, so that the scientific value of the end result will not be lost. By "measurement" we mean: (a.) a time-tagged value for the parameter; (b.) a corresponding uncertainty characterized sufficiently well to permit a probability density function for the error to be employed, e.g., a value for the standard deviation and a statement that the error is well approximated as Gaussian.

The Basic Algorithm

We will consider the case when two measurements with approximately Gaussian errors exist at times which bracket the epoch of interest. In such a case, the interpolation must be linear or something close to it. Generalization of the method presented below to a larger number of measurements is possible in various ways and will not be explored herein. The main concern will be with computing the uncertainty corresponding to the interpolated value. A simple method for doing this is described herein. Besides simplicity, the method has the following desirable properties:

> A.) If the values used for interpolation are essentially equal, suggesting that the parameter is constant, then the associated uncertainty reduces to that which would have been obtained by inverse-variance-weighted averaging of the two values, i.e., the uncertainty is smaller than either of the two measurement uncertainties.
>
> B.) If the values used for interpolation are slightly different, suggesting that the parameter drifts slowly, then the associated uncertainty inflates somewhat compared to case (A.) above, inflating most at interpolation points farthest from the measurements.

C.) If the values used for interpolation are *very* different, suggesting that the parameter drifts rapidly, then the associated uncertainty inflates by a large amount that is not unreasonable as a description of our ignorance about what is going on inside the interpolation interval; again the largest inflation is at interpolation points farthest from the measurements, and at the endpoints there is no reduction relative to the individual measurement uncertainties.

D.) The algorithm has surprisingly desirable extrapolation properties.

The method is simply a random-walk model of a Gaussian process, with the growth rate of the error variance estimated directly from the measurements themselves. Specifically, in keeping with the general absence of information, it is suggested that each measurement value be interpreted as a 50%-confidence random-walk excursion from the other, i.e., that what was observed is the least remarkable of all possible situations. Whereas a real calibration of the statistical properties behind the drift would be preferable to this approximation, the problem we are considering is one in which that is not possible, and the assumptions must reflect the lack of information. Fortunately, this approximation turns out to be less capricious than it might appear at first.

If there are theoretical qualms regarding this approximation, they probably stem from the initial appearance of a similarity to the assumption that the standard deviation of a population is about the same as the absolute difference of two samples drawn from the population, which is certainly a *very* non-robust estimate. But in the case herein, we are not using simple random draws; the two values whose difference is to determine an error-growth variance are themselves the results of *measurements*, so presumably some professional effort went into obtaining these values, and their uncertainties should be quite a bit smaller than they are. For example, the dark current of a pixel at one epoch is typically estimated by averaging many samples and rejecting outliers. Assuming typically small relative uncertainties, a given difference between dark estimates for that pixel at two epochs should be much more significant than the same difference between any two samples of its dark current at one epoch.

The important thing is whether this algorithm gives results that satisfy our intuition, so some examples will be given for the cases in which the two measurement values are equal, highly unequal, and in between. The method itself is a standard random walk model. Let the two measurement values of the calibration parameter be denoted M_i, i = 1, 2, with measurement variances V_i at epochs T_i defined such that $T_2 > T_1$. Then we define the time coefficient of the random-walk variance as follows.

$$V_{rw} = \frac{\left[\gamma\left(M_1 - M_2\right)\right]^2}{T_2 - T_1}$$

where γ is a coefficient that converts the observed difference from 50% confidence to one sigma (i.e., one standard deviation); for a Gaussian process, $\gamma = 1/0.6745 = 1.4826$. The error variance associated with measurement M_i at some time T is

$$V_{Ti} = V_i + V_{rw}\left|T - T_i\right|$$

The interpolated measurement M at time T is just the inverse-variance weighted mean of the two measurements, where the variances are the total variances, i.e., including the random walk from T_i to T:

$$V = \left(\frac{1}{V_{T1}} + \frac{1}{V_{T2}} \right)^{-1}$$

$$M = V \left(\frac{M_1}{V_{T1}} + \frac{M_2}{V_{T2}} \right)$$

where V is the error variance of M.

Note that the formula for V_{rw} attributes all of the difference between M_1 and M_2 to unexplained random walk, whereas some difference could be expected purely on the basis of the measurement uncertainties V_i. If one wished to remove the contribution of measurement error from the random-walk coefficient, one could use separate random walk coefficients for each measurement defined by

$$V_{rwi} = \frac{\left[\gamma \left(M_1 - M_2 \right) \right]^2 - V_i}{T_2 - T_1}$$

but then it would be necessary to clip negative values, and insisting on exactly compensating for the measurement uncertainties seems to be overinterpreting the formalism's rigor. Leaving the measurement error variances out of the random-walk coefficient could be interpreted as a conservative approximation or as the common practice of taking observed values to stand in for unknown true values. If one wishes to be more (or less) conservative, one may adjust the scale factor γ. For example, a value of 2 for γ corresponds to interpreting the observed difference as a half-sigma random walk, i.e., larger values of γ correspond to smaller fractions of random-walk sigma in the observed difference.

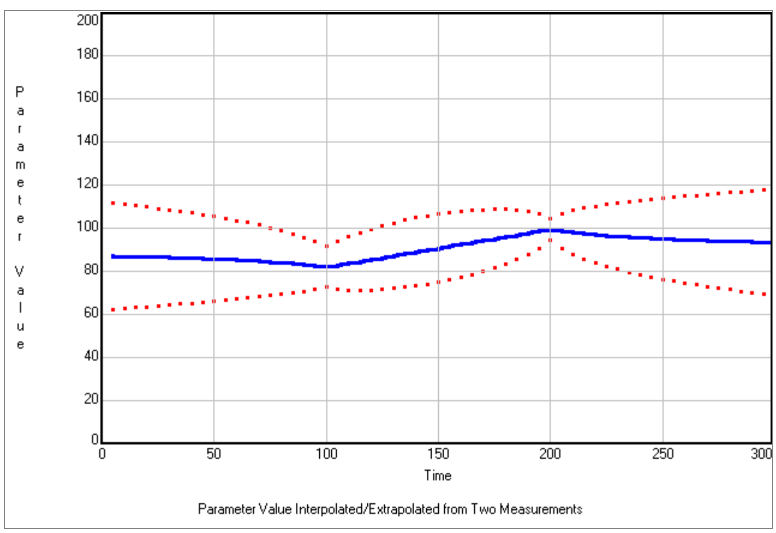

Figure 1

In the first example, $M_1 = 80$ and $M_2 = 100$ (in some convenient units), i.e., the parameter is drifting somewhat compared to the measurement errors, which are 10 and 5, respectively, one-sigma, and the epochs are at 100 and 200 (again, in some convenient unit). Figure 1 shows the interpolated value in blue; it turns out to be very close to a linear interpolation. The one-sigma envelope is shown in red dotted lines. We also allow the algorithm to extrapolate outside the measurement epochs to show what happens there: rather than extrapolate linearly, which would certainly violate the hypothesis that the parameter is supposed to be constant and simply drifts around somewhat unpredictably, the extrapolation approaches an unweighted mean asymptotically, because the average becomes dominated by the random-walk variances, which approach the same value far from the measurement epochs. But the uncertainty envelope does diverge, as indeed it should. This case corresponds to the situation for which this algorithm was invented: something which was expected not to change significantly did anyway, but not by absurd amounts. M_2 represents a 2-sigma deviation relative to M_1, which for a Gaussian error is about a 95%-confidence fluctuation, enough to cause some uneasiness but not to be considered critical. On the other hand, M_1 is a 4-sigma deviation relative to M_2, a bit more serious, but the real test is the difference divided by the root-sum-square uncertainty, which comes out 1.8 sigma. The conclusion should be that something is not quite right, but the situation can be salvaged.

The random-walk interpolation gives results close to the measurements for times near the measurement times. The worst-case interpolation occurs at $T = 145$, where $M = 90$ (the same as an unweighted mean), and the uncertainty is 15.8 (quite a bit more than either measurement, but probably still completely usable). If one is forced to extrapolate, M approaches the unweighted mean as the uncertainty diverges, placing a fairly natural limit on how far extrapolation can be pushed before the result becomes useless. In this case, with a bracket time interval of 100, at 100 units from the nearest measurement and outside the time bracket, the uncertainty is about 25. Over the entire range, these results appear much more reasonable than any nearest-in-time method, especially since such a method has no justifiable method for providing a plausible uncertainty.

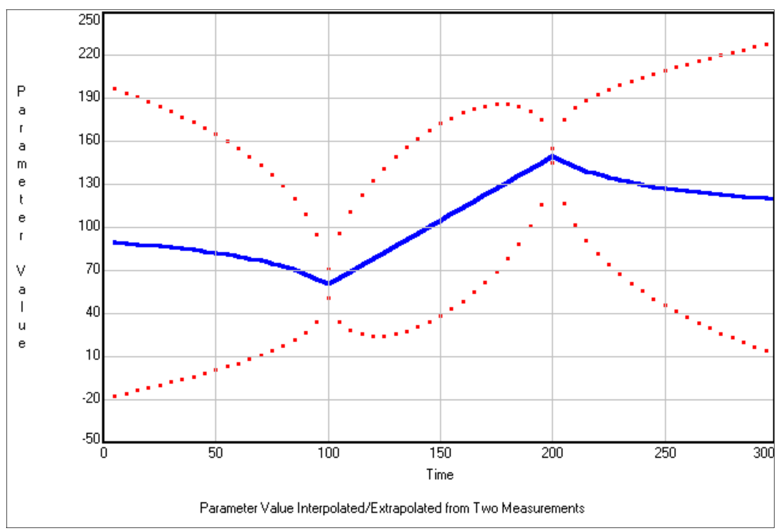

Figure 2

Figure 2 shows a more drastic case of drift. In this case, $M_1 = 50$ and $M_2 = 150$, with the same uncertainties as before. Here one is clearly on very thin ice and should consider a radical redesign, but if that is not possible, the results shown are probably the best one can do. Qualitatively, the behavior is the same as in the previous case, but now the uncertainties grow much larger for times not near measurements. At the extremes, the uncertainty is about 121. M takes on the value of the unweighted mean, 100, at $T = 150$, where the uncertainty is 74.3. Even fairly close to measurement times, the values are changing rapidly; for example, in going from $T = 100$ to $T = 105$, M has changed from 50 to 55.4, and the uncertainty has grown from 10 to 33.7. Given the measurements, there is no avoiding the great uncertainty in any interpolated values.

Finally we consider the case when no drift occurs; we have $M_1 = M_2 = 100$. This is shown in Figure 3. All interpolated and extrapolated results are constant, with $M = 100$ and the uncertainty is 4.47, i.e., a bit less than the smaller error bar of 5 but not much, since the error bar of 10 doesn't contribute much error reduction (if both uncertainties had been 5, the output uncertainty would have been 3.54, i.e., reduced by the square root of 2). On the one hand, the fact that the value has not changed could be taken as evidence that the two measurements should be combined into one refined estimate, which is what the algorithm provides. On the other hand, it may be felt that the parameter really does drift somewhat, and the nearly equal values are a coincidence. In the latter case, one could install a minimum-acceptable value for V_{rw}.

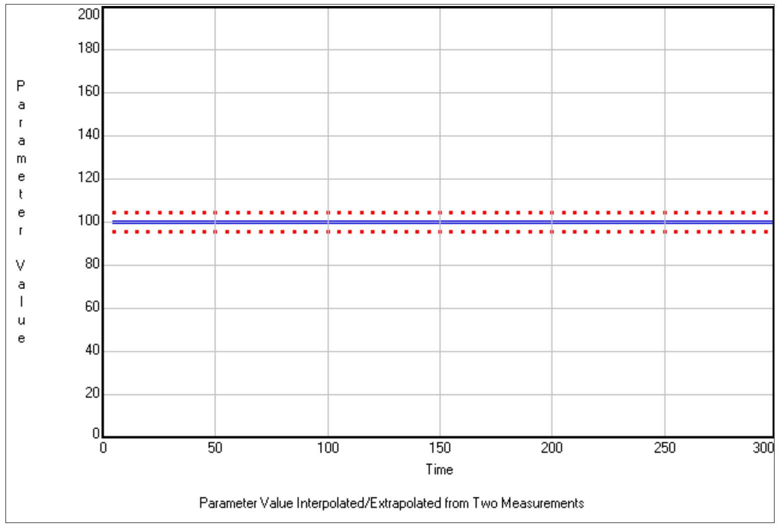

Figure 3

Extension of the Algorithm to Error Covariance Matrices

The discussion above can be applied to any parameter whose uncertainty is uncorrelated with that of other parameters, e.g., a pixel's dark current. In some cases, a set of parameters with uncertainties described by a full error covariance matrix must be handled, e.g., the coefficients of a linearity model. The off-diagonal elements of an error covariance matrix express the correlations in the errors of the parameters. Since nonzero correlations indicate additional information about the random variables, taking them into account always reduces the total uncertainty.

For example, since at least two random variables must be involved for correlation to be a relevant notion, consider the case of uncertainties in the celestial coordinates of a point source; then "total uncertainty" can be defined as the area inside a contour of constant joint probability density. The coordinates will be Right Ascension (α) and Declination (δ), the uncertainties will be expressed in true angular measure, and as before, Gaussian statistics will be assumed applicable. The last assumption results in constant-probability-density contours being elliptical (including the circular case), and we will use the contour that passes through the one-sigma points on the principal axes of the ellipse, which for zero correlation are the celestial coordinate axes. The area is therefore $\pi\sigma_\alpha\sigma_\delta$ in the uncorrelated case (note that although this example uses dimensionally identical parameters and therefore an

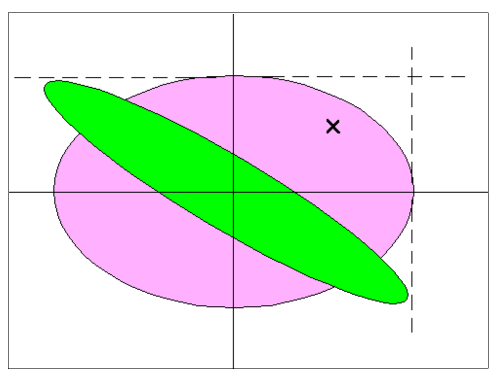

Figure 4. The error ellipse for correlated errors is rotated and has a smaller area than the error ellipse whose covariance matrix has the same diagonal elements but zero off-diagonal elements. Ignoring correlation would underestimate the significance of the deviation of the point marked by X.

intuitively obvious interpretation of "area", analogous interpretations apply to correlated random variables of different dimensionality, e.g., temperature and altitude, etc.).

When nonzero correlation exists, the elliptical contour's principal axes (say σ_a and σ_b for the major and minor axes, respectively) are no longer aligned with the coordinate axes, but the area inside the contour is still $\pi\sigma_a\sigma_b$. Figure 4 illustrates an error ellipse with strong correlation; this is the more eccentric ellipse that is rotated with respect to the coordinate axes. If we were to ignore the correlation, this would involve treating the off-diagonal elements of the covariance matrix as zero while keeping the diagonal elements as they are. The resulting error ellipse is the one shown as unrotated. When judging the significance of a deviation such as that marked as the point X, ignoring the correlation would result in an underestimate, as can be seen by the fact that the X is well within the unrotated ellipse but well outside the rotated one. It is obvious in the figure that it is also possible but much less likely to underestimate the significance of a deviation by ignoring correlation, i.e., there are small regions of the rotated ellipse that are farther from the origin than any part of the unrotated ellipse. Thus ignoring correlation does not always amount to being "conservative" in the sense of slightly overestimating uncertainties.

Two numerical examples will be given: one with equal standard deviations on the two axes and one with twice as much uncertainty in Right Ascension as in Declination. Various levels of correlation will be considered, all positive since negative values simply result in opposite rotations of the error ellipse. The error covariance in α and δ is denoted "cov(αδ)", and "Area" indicates the area inside the one-sigma contours.

Case 1: $\sigma_\alpha = 1$, $\sigma_\delta = 1$

cov(αδ)	Area
0	3.14159
0.1	3.12584
0.2	3.07812
0.3	2.99689
0.4	2.87932
0.5	2.72070
0.6	2.51327
0.7	2.24355
0.8	1.88496
0.9	1.36939
0.95	0.98096
0.99	0.44318
0.999	0.14046

Case 2: $\sigma_\alpha = 2$, $\sigma_\delta = 1$

cov(αδ)	Area
0	6.28319
0.1	6.27533
0.2	6.25169
0.3	6.21210
0.4	6.15624
0.5	6.08367
0.6	5.99378
0.7	5.88577
0.8	5.75863
0.9	5.61106
1.0	5.44140
1.5	4.15594
1.9	1.96192

Obviously the loss of correlation information would increase the total uncertainty. The purpose of the random-walk interpolation method is to take some account of the fact that we lose information as we consider epochs more and more distant from measurement times.

As an example of a case in which the random-walk interpolation algorithm must handle a full covariance matrix, we will consider a three-parameter linearity model. This involves a 3×3 matrix that expresses the uncertainty in the linear-fit coefficients. In this case the off-diagonal elements describe correlations in the uncertainties of the fit, and in the absence of any expectation that the drift in one parameter will depend on the value of another parameter, it will be assumed that the random walk affects only the diagonal elements. The algorithm generalizes to the following, where M is replaced by the vector $\boldsymbol{A}$ which contains the model coefficients a, b, and c, and V is replaced by a covariance matrix Ω. Subscripts will be used as above. The error covariance matrix for the random walk from epoch T_i to the time T is

$$\Omega_{RWi} \equiv \begin{pmatrix} \frac{\left[\gamma(a_1-a_2)\right]^2}{(T_2-T_1)} & 0 & 0 \\ 0 & \frac{\left[\gamma(b_1-b_2)\right]^2}{(T_2-T_1)} & 0 \\ 0 & 0 & \frac{\left[\gamma(c_1-c_2)\right]^2}{(T_2-T_1)} \end{pmatrix} \left|T-T_i\right|$$

$$\equiv \Omega_{RW} \left|T-T_i\right|$$

The error covariance matrix for the measurement at epoch i is

$$\Omega_i \equiv \begin{pmatrix} \sigma_a^2 & \mathrm{cov}(a,b) & \mathrm{cov}(a,c) \\ \mathrm{cov}(a,b) & \sigma_b^2 & \mathrm{cov}(b,c) \\ \mathrm{cov}(a,c) & \mathrm{cov}(b,c) & \sigma_c^2 \end{pmatrix}_i$$

The total uncertainty for the measurement at epoch i propagated to the time T is

$$\Omega_{Ti} = \Omega_{RWi} + \Omega_i$$

The "interpolated" vector $\boldsymbol{A}$ and its covariance matrix are

$$\Omega = \left(\Omega_{T1}^{-1} + \Omega_{T2}^{-1}\right)^{-1}$$
$$A = \Omega\left(\Omega_{T1}^{-1} A_1 + \Omega_{T2}^{-1} A_2\right)$$

where the subscripts on the $\boldsymbol{A}$ vectors refer to the two calibration epochs (not components of the vectors). This computation requires the inversion of symmetric real matrices. For the case of 3×3 matrices, inverses may be computed as follows. For a general real symmetric 3×3 matrix B

$$B \equiv \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{pmatrix}$$

the inverse is

$$B^{-1} = \frac{1}{D}\begin{pmatrix} b_{22}\,b_{33} - b_{23}^2 & b_{13}\,b_{23} - b_{12}\,b_{33} & b_{12}\,b_{23} - b_{13}\,b_{22} \\ b_{13}\,b_{23} - b_{12}\,b_{33} & b_{11}\,b_{33} - b_{13}^2 & b_{12}\,b_{13} - b_{11}\,b_{23} \\ b_{12}\,b_{23} - b_{13}\,b_{22} & b_{12}\,b_{13} - b_{11}\,b_{23} & b_{11}\,b_{22} - b_{12}^2 \end{pmatrix}$$

$$D \equiv b_{11}\,b_{22}\,b_{33} - b_{11}\,b_{23}^2 - b_{33}\,b_{12}^2 + 2\,b_{12}\,b_{13}\,b_{23} - b_{22}\,b_{13}^2$$

When coded on a computer, the algorithm should test for $D \le 0$ and execute a fatal-error termination if found. The determinant D is proportional to the square of the area inside the one-sigma contour of the error ellipse and can never be less than or equal to zero with realistic data, so the condition would indicate data corruption.

The two-dimensional case is similar to the three-dimensional case but with straightforward modification to one less dimension. This case is easier to visualize, so we will consider once again the problem of correlated errors in Right Ascension and Declination, denoted α and δ, measured in arbitrary units. The first measurement at $T = 100$ will have $\alpha = \delta = 100$, and the second at $T = 200$ will have $\alpha = 110$ and $\delta = 90$. For simplicity, let the two measurements have identical error ellipses with major axes of 10, minor axes of 2, and positive correlation that rotates the ellipses by 30 degrees. The actual error covariance matrices at the measurement times have the numerical values

$$\begin{pmatrix} 76.0 & 41.5692 \\ 41.5692 & 28.0 \end{pmatrix}$$

where the one-sigma principal axes of 10 and 2 become variances with values 100 and 4 before the 30-degree rotation. We compute the random-walk covariance matrix as prescribed above (except in two dimensions instead of three), and then we can evaluate the total covariance matrix as a function of time. Figure 5 shows the error ellipses (in sigma units rather than variance units) for times from 100 to 150 in steps of 5. At $T = 100$, a measurement time, the error ellipse is the smallest and most eccentric of the ellipses shown. Each larger ellipse corresponds to an interpolation time farther from the measurement until we arrive at the largest ellipse halfway between the measurements, after which the sequence reverses as we approach the other measurement. The ellipses are shown

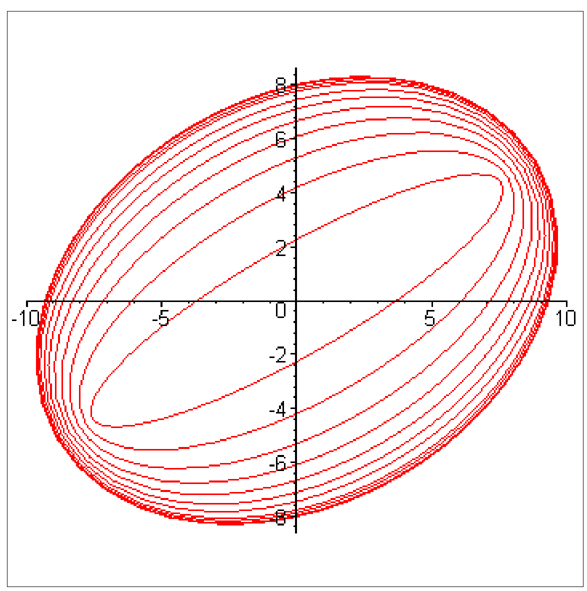

Figure 5. One-sigma error ellipses for times from 100 to 150 in steps of 5; the smallest ellipse occurs at the measurement time, T = 100, and the largest halfway between measurements, T = 150.

concentrically so that they may be compared to each other, but in the measurement space, each would be centered on the corresponding interpolated value of α and δ.

This illustration assumes that the position of an object on the sky changed significantly between two observations. Often such cases involve some knowledge about plausible proper motion, but here we are considering how to handle a situation that completely lacks such knowledge, and so we arrive at a nearly linear trajectory with error ellipses that are noticeably larger and less eccentric surrounding intermediate position estimates but which collapse almost exactly to the correlated error ellipses for times very close to the measurements.

For convenience we note here that if B had been a real symmetric 2×2 matrix such as in the two-dimensional example above,

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{12} & b_{22} \end{pmatrix}$$

its inverse would be

$$B^{-1} = \frac{1}{D}\begin{pmatrix} b_{22} & -b_{12} \\ -b_{12} & b_{11} \end{pmatrix}$$

$$D \equiv b_{11}\, b_{22} - b_{12}^2$$

Application to Spitzer Data, Part I: Nominal FITS Data

This algorithm is intended for implementation in the Spitzer module *caltrans* (*cal*ibration *trans*fer), which operates as a server of calibration data of various types for a requested epoch. Each combination of instrument, channel, observing mode, and calibration type (e.g., IRAC channel 1 sub-array mode flat field) must be separately bookkept by caltrans via its own data base. Here we consider a generic case involving FITS files. Besides a "primary" image (e.g., an IRAC array image) and its corresponding uncertainty image, there is also a *mask* image with the same dimensions. Spitzer mask images are bit-mapped two-byte integer images whose pixels correspond to the primary image pixels and whose individual bits are associated with certain documented conditions; if a bit is on, that condition is in effect in the primary and/or uncertainty image (e.g., if a pixel's bit number 9 is on, then a radiation hit has been diagnosed in the primary image's corresponding pixel).

Masks permit a lot of information to be tracked in a condensed space. Most processing modules use the mask image with a "fatal bit mask" parameter to control the processing. The simplest example is simply to skip a pixel for which the bitwise AND of the mask value and the fatal bit mask is nonzero. Herein we will simply refer to a pixel's mask being "OK" or "not OK" for processing. For brevity, we will also use this phrase to include the possibility of the primary or uncertainty pixel being a NaN, since that precludes applying the algorithm (and generally should be accompanied with a mask value that would be interpreted as fatal, but this cannot be guaranteed). Finally, a pixel's uncertainty value must not be zero; if it is, the pixel should also be treated as "not OK" for processing. In summary, caltrans must treat a pixel as masked if any of the following conditions are true:

1.) the bitwise AND of the mask value and the fatal bit mask is nonzero;
2.) the primary value or the uncertainty value is NaN, or both;
3.) the uncertainty value is zero.

FITS files must be processed by caltrans in three passes:

1.) All pixels that are OK at both epochs are processed as described above; the output mask value is the bitwise OR of the two input mask values; the mean value of V_{RW} (for single pixels) or Ω_{RW} (for parameter sets with covariance matrices) is computed; the standard deviation is also computed for QA purposes.

2.) All pixels that are OK at one and only one epoch are processed as described in the next section.

3.) All pixels that are not OK at either epoch are processed as described in the section after next.

Application to Spitzer Data, Part II: Partial FITS Data

This section describes the processing of FITS pixels that are OK at one and only one epoch. The primary output is the single good primary value M_k or parameter set A_k, where the index k indicates the usable epoch. The mask output is the same as the input but with a bit set, where the bit number is specified by the user. The uncertainty variance for a single pixel is

$$V_T = V_k + \overline{V_{rw}}\left|T - T_k\right|$$

where the coefficient of the absolute time difference is the mean random-walk coefficient computed for pixels with two usable epochs in the first pass through the data. Similarly, for parameter sets, the error covariance matrix is

$$\Omega_T = \Omega_k + \overline{\Omega_{rw}}\left|T - T_k\right|$$

In both cases, caltrans should have two CDF (controlled data file) values available: (a.) a canonical random-walk coefficient for use when no pixels have both epochs OK; (b.) a maximum output uncertainty. Both of these should be relative uncertainties. For case (a.), the parameter must be in the form of a relative random-walk variance per unit time, e.g., 4.0e-6/day would indicate that the relative variance due to drift increases by this amount per day; for a 100 day interval, the relative variance would be 4.0e-4, hence the relative sigma would be 0.02, or 2%, to be root-sum-squared with the measurement uncertainty. For case (b.), the computed uncertainty would be compared to that limit and clipped if necessary at that value, e.g., clip when the uncertainty (sigma) exceeds 50% of the output primary value. These two parameters could be bookkept for each combination of instrument, channel, observing mode, and calibration type, but it would probably suffice to use a global value if the instrument support team does not object.

Application to Spitzer Data, Part III: No Usable FITS Data

It may happen that for a given pixel, neither epoch contains usable data. This could happen because of the mask values or the primary values or the uncertainty values. We will consider the following cases.

1.) Neither primary value is NaN:
   a.) neither uncertainty value is NaN or zero: use the nominal algorithm to compute the primary and uncertainty output;
   b.) one and only one epoch has an uncertainty that is neither NaN nor zero: use the partial-data algorithm with this epoch to compute the primary and uncertainty output;
   c.) otherwise output the unweighted average primary value and the CDF maximum uncertainty value.

2.) One and only one epoch has a primary value that is not NaN:
   a.) the epoch's uncertainty is neither NaN nor zero: use the partial-data algorithm with this epoch to compute the primary and uncertainty output;
   b.) otherwise output the primary value and the CDF maximum uncertainty value.

3.) Both primary values are NaN: output NaN for primary and uncertainty.

In all cases, the mask output is the bitwise OR of the input mask values with one special bit set whose bit number is specified by the user. In cases where non-NaN output primary and uncertainty values are possible, the masks still contain fatal bits, so while the non-NaN values may be less reliable than usual, the mask values still provide protection, but the non-NaN values carry at least some information.

One final catastrophic situation is possible: no pixels are nominal, so no average random-walk coefficient can be computed. Pixels must be processed as in case1c, 2b, or 3 above, as appropriate.

Application to Spitzer Data, Part IV: Table-File Data

Caltrans also processes table-file data, which are different from FITS data in that no average random-walk coefficient can be computed from an ensemble of similar parameters, and there is no mask value to be considered. For an epoch to be unusable, either the primary or uncertainty value or both must be NaN, or the uncertainty value must be zero. For table-file data, this defines "not OK".

If both epochs are OK, the standard algorithm is used. If neither epoch is OK, only NaNs can be returned for primary and uncertainty values. If one and only one epoch is OK, then the primary value is returned, and the uncertainty is computed as described for partial FITS data using CDF values for the average random-walk coefficient and maximum uncertainty value.