

1 Extended WLSQM: dealing with missing function values

Can we extend WLSQM to the case where the function value $\hat{f}_i(x_i)$ is unknown, provided that $\hat{f}_i(x_k)$ is known for all neighbor points x_k ?

- The primary use case is handling boundary conditions, which may prescribe a derivative, leaving the function value free. In these cases, we eliminate the appropriate a_j , either by algebraic elimination of the corresponding row and column; or by replacing its row in the equation system with $1 \cdot a_j = C$ (maybe appropriately scaled), where C is its known value.
- This can also be used for interpolation, to obtain an approximation to the function value and its derivatives at an arbitrary point x that does not belong to the point cloud. (But here a cheaper alternative is to compute the approximation from the obtained quadratic fit. This also gives the derivatives, since the analytical expression of the fit is known.)
- Another use case may be as an error indicator (compare the interpolated $\hat{f}_i(x_i)$, computed by omitting f_i , and the actual data f_i).
- Also as a smoother? Replace each f_i by its interpolant, then iterate until convergence.

The answer turns out to be yes. Let us denote the local representation of our scalar field $f(x)$, in a neighborhood of the point x_i , by $\hat{f}_i(x)$.

Let us Taylor expand \hat{f}_i around the point x_i , and evaluate the Taylor series at a neighbor point x_k (a point distinct from x_i , also belonging to the point cloud):

$$\hat{f}_i(x_k) = \hat{f}_i(x_i) + h_k a_1 + \ell_k a_2 + \frac{h_k^2}{2} a_3 + h_k \ell_k a_4 + \frac{\ell_k^2}{2} a_5 + O(h_k^3, \ell_k^3), \quad (1)$$

where

$$h_k := (x_k)_1 - (x_i)_1, \quad (2)$$

$$\ell_k := (x_k)_2 - (x_i)_2, \quad (3)$$

and the function value and the derivatives are denoted by (note the numbering)

$$\begin{aligned} a_1 &= \frac{\partial \hat{f}_i}{\partial x} \Big|_{x=x_i}, & a_2 &= \frac{\partial \hat{f}_i}{\partial y} \Big|_{x=x_i}, \\ a_3 &= \frac{\partial^2 \hat{f}_i}{\partial x^2} \Big|_{x=x_i}, & a_5 &= \frac{\partial^2 \hat{f}_i}{\partial y^2} \Big|_{x=x_i}, \\ a_4 &= \frac{\partial^2 \hat{f}_i}{\partial x \partial y} \Big|_{x=x_i}, & a_0 &= \hat{f}_i \Big|_{x=x_i}. \end{aligned} \quad (4)$$

Truncating the error term, we have the Taylor approximation:

$$\hat{f}_i(x_k) \approx a_0 + h_k a_1 + \ell_k a_2 + \frac{h_k^2}{2} a_3 + h_k \ell_k a_4 + \frac{\ell_k^2}{2} a_5 =: \bar{f}_k, \quad (5)$$

Now, let us define the coefficients

$$\begin{aligned} c_k^{(1)} &:= h_k, & c_k^{(2)} &:= \ell_k, \\ c_k^{(3)} &:= \frac{h_k^2}{2}, & c_k^{(5)} &:= \frac{\ell_k^2}{2}, \\ c_k^{(4)} &:= h_k \ell_k, & c_k^{(0)} &:= 1. \end{aligned} \quad (6)$$

Observe that

$$\frac{\partial \bar{f}_k}{\partial a_j} = c_k^{(j)}, \quad (7)$$

At the neighbor points x_k (belonging to the point cloud), by assumption we have the function values available as data. The error made at any such point x_k , when we replace $\hat{f}_i(x_k)$ with its Taylor approximation, is

$$e_k := f_k - \bar{f}_k, \quad (8)$$

One-half of the total squared error across all the neighbor points k is simply

$$G(a_0, \dots, a_5) := \frac{1}{2} \sum_{k \in I_i} e_k^2, \quad (9)$$

where I_i is the index set of the point i 's neighbors.

Minimizing the error leads, in the least-squares sense, to the best possible values for the a_j :

$$\{a_0, \dots, a_5\}_{\text{optimal}} = \arg \min_{a_0, \dots, a_5} G(a_0, \dots, a_5).$$

Because $G \geq 0$ for any values of the a_j , and G is a quadratic function of the a_j , it has a unique extremal point, which is a minimum. The least-squares fit is given by this unique minimum of G :

$$\frac{\partial G}{\partial a_j} = 0, \quad j = 0, \dots, 5. \quad (10)$$

Using first (9), and then on the second line (8), we can write

$$\begin{aligned} \frac{\partial G}{\partial a_j} &= \sum_{k \in I_i} e_k \frac{\partial e_k}{\partial a_j} \\ &= \sum_{k \in I_i} [f_k - \bar{f}_k(a_0, \dots, a_5)] \left[-\frac{\partial \bar{f}_k}{\partial a_j} \right] = 0, \quad j = 0, \dots, 5, \end{aligned} \quad (11)$$

which, using (5)–(6) and (7), leads to

$$\sum_{k \in I_i} \left([-f_k + c_k^{(0)} a_0 + c_k^{(1)} a_1 + c_k^{(2)} a_2 + c_k^{(3)} a_3 + c_k^{(4)} a_4 + c_k^{(5)} a_5] c_k^{(j)} \right) = 0, \quad j = 0, \dots, 5.$$

This can be written as a standard linear equation system

$$\sum_{n=0}^5 A_{jn} a_n = b_j, \quad j = 0, \dots, 5, \quad (12)$$

where

$$A_{jn} = \sum_{k \in I_i} c_k^{(n)} c_k^{(j)}, \quad (13)$$

$$b_j = \sum_{k \in I_i} f_k c_k^{(j)}. \quad (14)$$

Considering the magnitudes of the expressions (6), which contribute quadratically to A_{jn} , we see that the condition number of A will likely deteriorate, when compared to the previous case where f_i is known. This is as expected; we are now dealing with not only the first and second derivatives, but also with the function value. Preconditioning (even by row normalization, although this destroys the symmetry) may help with floating-point roundoff issues.

2 Accuracy?

At first glance, WLSQM is not even consistent, if we treat \bar{f}_k as an $O(h_k^3, \ell_k^3)$ approximation of $\hat{f}_i(x_k)$, and track the error term through the calculation (details left as an exercise).

Obviously, consistent expansion of the matrix A to the order $O(h_k^3, \ell_k^3)$ gives

$$A = \sum_{k \in I_i} \begin{bmatrix} 1 & h_k & \ell_k & \frac{h_k^2}{2} & h_k \ell_k & \frac{\ell_k^2}{2} \\ h_k & h_k^2 & h_k \ell_k & \sim 0 & \sim 0 & \sim 0 \\ \ell_k & h_k \ell_k & \ell_k^2 & \sim 0 & \sim 0 & \sim 0 \\ \frac{h_k^2}{2} & \sim 0 & \sim 0 & \sim 0 & \sim 0 & \sim 0 \\ h_k \ell_k & \sim 0 & \sim 0 & \sim 0 & \sim 0 & \sim 0 \\ \frac{\ell_k^2}{2} & \sim 0 & \sim 0 & \sim 0 & \sim 0 & \sim 0 \end{bmatrix} = \sum_{k \in I_i} \begin{bmatrix} 1 & h_k & \ell_k & \frac{h_k^2}{2} & h_k \ell_k & \frac{\ell_k^2}{2} \\ & h_k^2 & h_k \ell_k & \sim 0 & \sim 0 & \sim 0 \\ & & \ell_k^2 & \sim 0 & \sim 0 & \sim 0 \\ & & & \sim 0 & \sim 0 & \sim 0 \\ \text{symm.} & & & & \sim 0 & \sim 0 \\ & & & & & \sim 0 \end{bmatrix}$$

which has at most rank 3 (rank 2 in the classical case, where the first row and column, corresponding to unknown f_i , are removed). Of course it can be of full rank if the almost zeros are retained, but the truncation error (of the Taylor approximation) dominates those, so consistency requires that they be dropped.

(As an aside, we note that if there is only one neighbor point, the equations corresponding to the UL 3x3 block become scalar multiples of each other due to b also having a factor of $c_k^{(j)}$. This is of course as expected; one can hardly expect to obtain two independent derivatives from just one neighbor. The same occurs if the neighbors are collinear (as is obvious geometrically, and quite simple to see algebraically, writing e.g. for two points $h_2 = Ch_1$, $\ell_2 = C\ell_1$...).

However, WLSQM (at least the classical version with f_i known) has been observed to actually work, with some reasonable amount of numerical error, so this analysis must be wrong. What is going on?

2.1 Accuracy, correctly

Let us take a page from finite element methods, where the weak form is — after the fact — taken as the new *definition* of the problem (which just so happens to lead to the classical strong form in cases where both can be written).

To apply this philosophy here: after we define \bar{f}_k , we can “forget” that it comes from a truncated Taylor series, and *take the definition as a new starting point*: in principle, \bar{f}_k is just a function of the a_j , to be least-squares fitted to known data points f_k (and optionally known f_i , as per classical WLSQM).

Then we just perform standard least-squares fitting. The math is exact (given unrealistic, exact arithmetic — this is a separate issue); no truncation error term appears. The full matrix should be retained:

$$A = \sum_{k \in I_i} \begin{bmatrix} 1 & h_k & \ell_k & \frac{h_k^2}{2} & h_k \ell_k & \frac{\ell_k^2}{2} \\ h_k & h_k^2 & h_k \ell_k & \frac{h_k^3}{2} & h_k^2 \ell_k & h_k \frac{\ell_k^2}{2} \\ \ell_k & h_k \ell_k & \ell_k^2 & \frac{h_k^2 \ell_k}{2} & h_k \ell_k^2 & \frac{\ell_k^3}{2} \\ \frac{h_k^2}{2} & \frac{h_k^3}{2} & \frac{h_k^2 \ell_k}{2} & \frac{h_k^4}{4} & \frac{h_k^3 \ell_k}{2} & \frac{h_k^2 \ell_k^2}{4} \\ h_k \ell_k & h_k^2 \ell_k & h_k \ell_k^2 & \frac{h_k^3 \ell_k}{2} & h_k^2 \ell_k^2 & h_k \frac{\ell_k^3}{2} \\ \frac{\ell_k^2}{2} & \frac{h_k \ell_k^2}{2} & \frac{\ell_k^3}{2} & \frac{h_k^2 \ell_k^2}{4} & h_k \frac{\ell_k^3}{2} & \frac{\ell_k^4}{4} \end{bmatrix} = \sum_{k \in I_i} \begin{bmatrix} 1 & h_k & \ell_k & \frac{h_k^2}{2} & h_k \ell_k & \frac{\ell_k^2}{2} \\ & h_k^2 & h_k \ell_k & \frac{h_k^3}{2} & h_k^2 \ell_k & h_k \frac{\ell_k^2}{2} \\ & & \ell_k^2 & \frac{h_k^2 \ell_k}{2} & h_k \ell_k^2 & \frac{\ell_k^3}{2} \\ & & & \frac{h_k^4}{4} & \frac{h_k^3 \ell_k}{2} & \frac{h_k^2 \ell_k^2}{4} \\ \text{symm.} & & & & h_k^2 \ell_k^2 & h_k \frac{\ell_k^3}{2} \\ & & & & & \frac{\ell_k^4}{4} \end{bmatrix}$$

This is now of full rank, provided that enough neighbor points x_k are used in the calculation (considering that we are least-squares fitting a general quadratic polynomial in the plane; see below).

At this point the only error — considering only \bar{f}_k and the data f_k — is the *RMS (root mean square) error* of the least-squares fit, $\min \sqrt{2G}$ (where the minimum occurs at the solution point). The RMS error measures how well the model adheres to each data point, on average. The obtained coefficients are optimal: out of all functions of the form (5) with a_j as parameters, the solution of (12)–(14) gives the smallest possible RMS error for the fit.

Then — again after the fact — we observe that these optimal a_j are pretty good also for use in a Taylor approximation. The solution is, in the least-squares sense, the best quadratic polynomial of (x, y) for locally approximating $f(x)$ around x_i . (The fit (5) is linear in a_j , but quadratic in (x, y) .) Also the Taylor approximation, truncated after the second-order terms, is a quadratic polynomial approximating $f(x)$ around x_i . Thus *we interpret the quadratic fit as a (response surface) model* for $f(x)$ near x_i , and thus the a_j as approximations to the Taylor coefficients of f (whence also as the numerical approximations to the derivatives).

However, it must be emphasized that this gives rise to *modeling error*, because the a_j are *not* the exact coefficients of the true Taylor expansion of $f(x)$ around x_i . Indeed, strictly speaking, the data may not even describe a function admitting such an expansion! Even if the data admits an underlying function, and it happens to be in C^2 , there may be numerical and/or experimental noise in the data points, depending on the data source. (This *inexact data* is another separate error source.) Also, in the general case the fit will not be exact, i.e. the RMS error will be nonzero.

From this viewpoint, WLSQM would be more accurately advertised as a method for response surface modeling (RSM), for computing a local quadratic response surface in arbitrary geometries, instead of as a method for numerical differentiation.

Regarding numerical differentiation, the natural follow-up question is, what is the total error arising from approximating the function $f(x)$ locally as the quadratic polynomial fit? The (original, not truncated) Taylor series, at a general point x in the neighborhood of x_i , is

$$\hat{f}_i(x) = \hat{f}_i + h \frac{\partial \hat{f}_i}{\partial x} + \ell \frac{\partial \hat{f}_i}{\partial y} + \frac{h^2}{2} \frac{\partial^2 \hat{f}_i}{\partial x^2} + h\ell \frac{\partial^2 \hat{f}_i}{\partial x \partial y} + \frac{\ell^2}{2} \frac{\partial^2 \hat{f}_i}{\partial y^2} + O(h^3, \ell^3), \quad (15)$$

where on the right-hand side, the function and the derivatives are evaluated at $x = x_i$, and $x - x_i = (h, \ell)$. The quadratic polynomial fit is

$$Q(x) := a_0 + ha_1 + \ell a_2 + \frac{h^2}{2} a_3 + h\ell a_4 + \frac{\ell^2}{2} a_5, \quad (16)$$

where the a_j are obtained from the least-squares optimization. The total error in the function value, at a point x , is their difference

$$\begin{aligned} \text{err}(x) &:= f(x) - Q(x) \stackrel{\text{near } x_i}{=} \hat{f}_i(x) - Q(x) \\ &= (\hat{f}_i - a_0) + h\left(\frac{\partial \hat{f}_i}{\partial x} - a_1\right) + \ell\left(\frac{\partial \hat{f}_i}{\partial y} - a_2\right) + \frac{h^2}{2}\left(\frac{\partial^2 \hat{f}_i}{\partial x^2} - a_3\right) + h\ell\left(\frac{\partial^2 \hat{f}_i}{\partial x \partial y} - a_4\right) + \frac{\ell^2}{2}\left(\frac{\partial^2 \hat{f}_i}{\partial y^2} - a_5\right) + O(h^3, \ell^3). \end{aligned} \quad (17)$$

When the Taylor series is truncated after the quadratic terms, the asymptotic term gives the *truncation error*. The rest of the error is due to *modeling error* in the coefficients a_j , i.e. the parenthetical expressions in (17).

It is obvious that in the general case, the modeling error will be nonzero (even if we assume the data to be exact): the function f is generally not a quadratic polynomial, and hence no quadratic polynomial can represent it exactly. To reiterate: the coefficients a_j are the coefficients of the quadratic fit $Q(x)$ — they are *not* the Taylor coefficients of f !

However, they are a computable, close relative of the Taylor coefficients of the unknown function f , since the Taylor series of $Q(x)$ expanded at x_i is, quite simply, $Q(x)$ itself. (Because $Q(x)$ is a polynomial, no asymptotic error term appears.)

Thus the magnitude of the total error depends on how well the coefficients a_j approximate the Taylor coefficients of f ; or in other words, how close f is (locally) to a quadratic polynomial (which — given exact data and exact arithmetic — can be fitted exactly; note that both assumptions are required, as inexact data will give rise to nonzero RMS error in the fit, i.e. then the fit will not be exact).

This obviously depends on the neighborhood size, due to the asymptotic term describing the truncation error in (15). The asymptotic term of the Taylor series says that if the neighborhood is small enough, any $f \in C^2$ is locally close to a quadratic polynomial. This — for sufficiently small neighborhoods — should make the modeling error (and thus the error in the numerical derivatives) comparable to $O(h^3, \ell^3)$.

This suggests that $\text{err}(x)$ — with exact data and exact arithmetic — should also be comparable to $O(h^3, \ell^3)$. (With inexact data, one needs to take into account that $f_k = f(x_k) + \delta_k$ and work from that.)

Observe also that there are six a_j ($j = 0, \dots, 5$) in (16). Hence, with exact arithmetic, six data values for (5), i.e. six neighbors x_k (five if f_i is known, eliminating a_0), uniquely determine the quadratic function $Q(x)$. (Fewer data values lead to an underdetermined system, which has an infinite family of solutions.) More data values lead to an overdetermined system, which is then taken care of by least-squares fitting: picking the quadratic polynomial that best approximates the data (which generally did not come from a quadratic polynomial).

This explains why the classical WLSQM takes 6 neighbors (here 7 if f_i is not known) to perform the fitting; it is the smallest number of (nondegenerate!) neighbors x_k that makes the quadratic fitting problem overdetermined (hence actually needing the least-squares procedure). (The overdeterminedness also slightly protects against inexact data, so that one data point that is slightly off will not completely change the fit.)

But why is the result not exact, i.e. why is there modeling error in (17)? After all, the truncated Taylor expansion *is* the best local polynomial representation of f , up to the given degree. With exact data and arithmetic, how can the least-squares fit be anything but the truncated Taylor expansion?

The key is in the definition of “best”. In a Taylor series, as the truncation order is increased, with each added term (given sufficient continuity of f) the asymptotic accuracy increases, without requiring changes to the already computed coefficients. The Taylor series, being the polynomial series expansion of f , is optimal in the class of polynomial representations where the coefficients are “final” in this sense. This is indeed what leads to the common-sense notion of the Taylor series being “the best polynomial representation” of f .

However, nothing requires the truncated Taylor series to satisfy the least-squares property. In the least-squares sense, *there may exist better polynomials of the same degree* to locally approximate f . For a trivial 1D example: to represent $f(x) = x^2$ in an interval $x \in [-a, a]$ around the origin using a constant approximation, the Taylor series produces $f \approx 0$. However, the mean value across the interval is a “better” constant approximation in an integral least-squares sense.

Indeed, a least-squares fit, as its order is increased, will change *all* of its coefficients; and it will do this to minimize the RMS error of the fit. (Be very careful: this is different from the modeling error in equation (17). The RMS error only measures how well the model adheres to each data point, on average; it does not see what the model is used for.) In the least-squares fit, there is no asymptotic error term — the data points f_k , used in the fitting, implicitly contain the information also from all the higher-order terms in the polynomial series expansion of f . The fit then eliminates as much of the difference between the chosen model and the data as is possible.

It is not surprising that the price that must be paid for this increase of accuracy in interpolation is the “finality” of the coefficients in the above sense, the Taylor series already being optimal in its class.

We conclude that in the general case the result cannot be exact, because we are dealing with two very different entities, which coincide only under very restrictive assumptions.

Note also that “best” obviously depends on context. For response surface modeling, the WLSQM quadratic polynomial fit is optimal. However, for numerical differentiation, the fact that the obtained coefficients do not exactly coincide with the Taylor series coefficients of f produces an undesirable source of numerical error (modeling error).