Modeling Velocity Gradients in an OBC, First-Break Positioning Algorithm

Noel Zinn, Western Geophysical, 3600 Briarpark Drive, Houston, Texas 77042, USA

European Association of Geoscientists & Engineers, Geneva, May 26-30, 1997

Introduction

Consider an Ocean Bottom Cable (OBC) prospect. A subset of one swath of such a prospect is shown in Figure 1, an array of just 16 detectors (geophysical receivers, which, in this case, happen to be coupled with acoustic sensors) and 2533 orthogonally-fired shots. Depending on offset and water depth, the onset of seismic energy may arrive via the water column or through one or more refractive layers. Typically, the velocity of refracted energy may vary vertically (with depth) either within one refractor or between successively deeper, faster refractors. Consider that in this prospect velocity may also vary laterally (in horizontal space) with, for example, velocities in the north-west being faster than in the south-east. This situation may be less typical, but it does often occur. A first break is a measure of the travel time of the onset of seismic energy from a source to a detector. Because sources and detectors are distributed widely in a pattern throughout the prospect, travel paths may vary in offset (distance between source and detector) and in horizontal space (from north to south, north to north, south to south, and so on). The shallower the water, the more likely is the onset of energy to be refracted. The longer the offset, the more likely a deeper, faster refractor may be involved. The more north-easterly the travel path (in this example), the faster the velocity will be. Ignoring the effects of vertical and lateral velocity gradients (that is, assuming travel times are linearly related to distance traveled) will introduce biases in coordinates computed with a simple, least-squares, first-break positioning algorithm. This paper describes two methods of dealing successfully with these and other problems.

Ocean-Bottom-Cable Positioning

Due to various geophysical advantages (not discussed in this paper), OBC seismic surveys are gaining popularity in water depths down to 150 meters and perhaps more. Source positioning in OBC is similar in technique and quality to source positioning in deep-water streamer surveys. It basically consists of Global Positioning System (GPS) receivers on the source array. On the other hand, detector positioning techniques are less-widely standardized in OBC than in land or deep-water streamer surveys. Three techniques are common in the industry: (1) recording and using the drop coordinates of the detectors, (2) deploying high-frequency acoustic sensors attached to all or some of the detectors and positioned by a "pinging" survey independent of the seismic survey and (3) using multiple occasions of the onset of seismic energy (first breaks) as surveying observations in a positioning algorithm. A combination of first breaks and acoustics is also possible.

Advantages and Disadvantages

Since drop positions must be recorded anyway to assure that the actual detector location bears some resemblance to the planned location, this technique is the cheapest and easiest to implement. In shallow water the detector drop position can be close to the resting position. But in deeper water this is not always likely due to currents and drop trajectories. On the other hand, ancillary, high-frequency acoustic systems are provided by a number of vendors. Acoustic surveys can be quite accurate within the limitations of known detector depth, known velocity of acoustic propagation in water (especially considering thermal layering), an adequate number of pings in good geometry, multi-path (surface "ghosts") and signal reception (which can be obscured by vessel noise or muddy bottoms). Unfortunately, acoustic positioning is expensive (extra equipment) and operationally time consuming. Limiting the expense (e.g. interpolating detector coordinates between fewer acoustic sensors) or the time (e.g. pinging less) has a profound impact on the precision and reliability of detector coordinates when using acoustics. Alternatively, first-breaks can be picked by any number of automated methods that choose a significant change in the amplitude or inflection of the arriving, perhaps-preconditioned seismic energy. Automated picking can be enhanced with neural networks, or troublesome picks can be made by hand. The time of a first-break pick can be related to distance. Distances can be processed in a positioning algorithm. First-break positioning potentially combines the cost advantages of drop positions with the accuracy of acoustics. In a seismic survey, the marginal cost of picking and processing first breaks is low since the personnel, software and seismic data are already on the job to reposition the swath immediately after shooting. Although each first break is a crude observable (good to plus or minus 2-6 milliseconds or 3-9 meters, or worse, in a random sense), we enjoy an abundance of observations, especially when refracted energy is processed. (Limiting first-break picks to water arrivals severely limits the number of first-break observations, especially in shallow water.) Laws of statistical error cancellation in a large sample of random observations readily confirm that acoustic-quality results are possible with first breaks.

Systematic Errors

First-break errors are not only random, however. Systematic first-break errors may be caused by the geometry of the source array, instrumental delay in the seismic recorder, different definitions of the onset of energy in different picking algorithms, different velocities of propagation in the water and the several refractors through which the onset of energy may arrive (vertical velocity gradient), spatial variation in the refractors (lateral velocity gradient) and, of course, a complex near-surface geology (a catchall for other, less-common factors). A competent first-break positioning algorithm must have strategies for dealing with all these sources of systematic error.

Global Polynomial Vertical Velocity Profile

The geometry of the source array is easily compensated with simple, albeit tedious, programming since all the relevant factors are known. The use of a vertical velocity polynomial is an effective and automated way to deal simultaneously with instrumental delay, the definition of the onset of energy and the existence of a vertical velocity gradient. The polynomial fits all offsets to all pick times in the swath being processed before sending pick time distances to the adjustment algorithm. Offsets are derived by "inversing" GPS source and detector drop coordinates, that is, by determining the Pythagorean distance with a little analytical geometry. The offsets are revised upon each iteration of the algorithm as coordinates are refined. Setting the order of the polynomial to two plus the number of travel paths (water and one or more refractors) is generally effective. The zero-order coefficient of the polynomial absorbs instrumental delay and any delay in the definition of the onset of energy in the picker. The higher order terms model global vertical velocity of propagation as a function of offset.

Figure 1 shows the real-data swath subset used for illustration in this paper. Among the possible 40,528 picks among all combinations of sources and detectors, 23,547 picks fall within a 1,500-meter offset limitation from the drop positions of the detectors imposed for convenience and to balance the geometry in processing. Notice that this swath was shot around an obstruction, an excellent application of OBC. Figure 2 shows a plot of picks (the dots) in milliseconds on the X axis versus offset in meters on the Y axis before repositioning. Although offsets will change as the algorithm iterates, pick times are fixed. (Of course, re-picking with different automated methods or manual intervention is an option.) Figure 3 shows a well-fitting, 4th-order polynomial computed for these data. Notice the small offset at time zero that models both instrumental delay and the characteristic delay or anticipation of the picking method used. Notice also that pick times can be converted to pick time distances by reading them off the plot or, more efficiently, by substituting them into the polynomial, which takes the general form:

Eq 1 $$ptd = c_0 + c_1 \cdot pt + c_2 \cdot pt^2 + c_3 \cdot pt^3 + \ldots + c_n \cdot pt^n$$

where *ptd* is pick time distance in units of distance, *pt* is pick time in units of time and *n* is the order of the polynomial. The coefficients c_0 through c_n are estimated by least squares. Differentiating this polynomial with respect to time provides an equation for velocity as a function of pick time. This form of the polynomial is stable numerically up to about the fifth order, depending on various factors, and this is generally adequate for most geological conditions. Reformulating the equation with Chebyshev polynomials allows numerical stability up to higher orders with equivalent results (for equivalent orders), if desired. But differentiation in the Chebyshev case is more difficult to interpret.

Fitting these same data with a first-order polynomial (linear regression) and repositioning yields Figure 4, a plot of pick time distance residuals in meters (the dots) versus pick times in milliseconds. A residual is also known as the C-O, or the computed minus the observed. It is the finally-computed offset (as defined above) minus the pick time distance on the final iteration of the positioning algorithm. The trend as a function of pick time is obvious. Figure 5 is the equivalent residual plot using a fifth-order polynomial. Notice that the higher-order polynomial eliminates most of the trend and significantly reduces the average residual. Since predicted error (precision) is a function of the average residual, precision is also enhanced. Geodesists define predicted error as the error ellipse scaled by the variance factor.

Figure 5, which exhibits the benefit of modeling the global vertical velocity gradient, also has important implications for the efficacy of any outlier rejection scheme implemented in the processing algorithm. Because outlying picks (blunders, spikes, faulty data outside the normal, random population of acceptable picks) can pollute a least-squares algorithm, they must be rejected. It is easy to see in Figure 5 that a rejection tolerance of plus or minus 35-40 meters of residual will separate the inliers from the outliers. Imposing such a tolerance without the global vertical velocity model (as in Figure 4) will certainly lead to the rejection of good data. Real-world picks can be worse than those in this example. Of course, more sophisticated, statistically-based outlier detection schemes (such as the Delft method) can be implemented (ref. Cross).

Lateral Velocity Gradient

A lateral velocity gradient is a variation in velocity as a function of position in a geological field. Different than anisotropy, it may be uniform in all directions at a specific point. (A lateral velocity gradient behaves like scale factor in what cartographers refer to as a conformal map projection.) It may be caused, for example, by a greater compaction of sedimentation as one moves farther offshore. Since the refracted energy used in OBC first-break positioning primarily travels through the recent sedimentary layers, a lateral velocity gradient may sometimes be a factor in positioning results. A simple least-squares algorithm will give erroneous results in the presence of a lateral velocity gradient, with coordinates biased in the direction of the gradient.

A possible lateral velocity gradient must be modeled in a competent first-break pick positioning algorithm. One way is to compute source-specific and detector-specific velocity scalars in independent gathers by least-squares estimation and then to apply them to the pick time distances before positioning. Another is to rewrite the positioning observation equation to include spatial variation in velocity. The coefficients of the velocity model are then solved in a network adjustment. **Appendix A** details this more-rigorous approach and also discusses the computational benefits of Helmert blocking in this application. Figure 6 represents the lateral velocity gradient computed by this method for the same real data example as depicted in the previous figures. Notice the gradient from north-west to south-east, the direction of this swath subset. The peaks in the north-east and south-west are artifacts of extrapolating the quadratic surface into territory unconstrained by real data in this plot.

Helmert Algorithm

A flow diagram of the Helmert algorithm, which includes both vertical and lateral velocity gradient modeling, is given in Figure 7. Geometry permitting, the algorithm can be used to refine detector or source coordinates, or both. For illustration, I consider only detector positioning in this paper. The algorithm is iterated until the final coordinates differ from the coordinates on the last iteration by some defined convergence tolerance.

Numerical Comparisons

A first-break positioning algorithm employing vertical and lateral velocity-modeling techniques can be validated in three ways: (1) by the processing of simulated (synthetic) data, (2) by comparing real-data results with alternative systems (such as acoustics) and (3) by splitting the first-break data into independent samples either randomly over all offsets or by offset into nears and fars. Data splitting results are compared for the Helmert algorithm only in the two tables that follow. The positioning results from independent first-break samples chosen over the entire offset range compare exceedingly well, specifically within the predicted error of the algorithm. It is a greater challenge to get agreement between exclusive offset ranges, that is, between the nears and the fars. The comparisons that follow will concentrate on this kind of data splitting because it has profound implications for modeling a final error source mentioned in the conclusion of this paper. Note that data splitting compares results computed from completely independent samples of first-break picks, that is, no shared observations.

Figure 8 is a residual plot of data simulated with both a vertical and a lateral (north-south) velocity gradient and processed in a least-squares algorithm that does not model these gradients. A normally distributed random error of 4 milliseconds (one sigma) is added to the simulated picks. The picks are then rounded to the nearest 4 milliseconds to emulate a commonly-used sampling interval that produces excellent positioning results. Direct water arrivals and three distinct refractors of different thickness are programmed into the simulator. A linear lateral velocity gradient in the Y coordinate only was programmed into the simulator to modify the simulated picks. The effect of the vertical gradient is obvious in Figure 8; the lateral gradient cannot be seen in this plot. Since the velocity in each refractor is constant throughout its thickness in this version of the simulator, the refractor "breaks" are quite distinct. This sometimes occurs similarly in nature. By comparison, notice that the real data of Figure 4 indicates a velocity variation within its one or more refractors, that is, smoother transition over the offset range.

Table 1: Simulated-Data Results minus Truth

Algorithm	Offset Range	Mean ΔX	SD ΔX	Mean ΔY	SD ΔY
LS (no gradients)	0-1500 m	0.15 m	0.67 m	-13.32 m	0.58 m
LS (no gradients)	0-900 m	0.08 m	0.86 m	-4.63 m	0.64 m
LS (no gradients)	900-1500 m	0.16 m	0.65 m	-19.11 m	0.76 m
LS (vertical gradient only)	0-1500 m	0.15 m	0.68 m	-13.65 m	0.55 m
LS (vertical gradient only)	0-900 m	0.07 m	0.87 m	-4.63 m	0.65 m
LS (vertical gradient only)	900-1500 m	0.21 m	0.61 m	-18.73 m	0.76 m
Helmert (both gradients)	0-1500 m	0.21 m	0.47 m	0.16 m	0.53 m
Helmert (both gradients)	0-900 m	-0.23 m	0.73 m	0.05 m	0.60 m
Helmert (both gradients)	900-1500 m	0.08 m	0.85 m	-0.31 m	0.92 m
Helmert (both gradients)	fars - nears	0.31 m	1.19 m	-0.36 m	1.08 m

Table 1 gives the results of processing these simulated data with three algorithms: (1) Least Squares (LS) with no velocity gradients modeled, (2) Least Squares (LS) with just the vertical velocity gradient modeled and (3) Least Squares (Helmert) with both vertical and lateral velocities modeled and using the Helmert-blocking technique for computational efficiency as described in **Appendix A**. The comparisons in Table 1 are with the true coordinates of the simulator. Notice that offset ranges are in meters for the simulated picks since true coordinates are known. Notice that the results of algorithms (1) and (2) are comparable, that is, that the modeling of the vertical velocity gradient does not appear to offer any benefit. This result is largely a consequence of the balanced geometry of the detectors in the simulator. The distribution of sources is such that there is an even distribution of offsets over all azimuths, a consequence of the orthogonal shooting style programmed into the simulator. In-line shooting does not offer as much of this benefit and, consequently, modeling the vertical velocity gradient becomes more important. Notice that discrepancies with truth are negligible in the X coordinate, but significant in the Y coordinate for the LS algorithms without a lateral velocity model. All of the linear gradient was programmed into the Y coordinate and that's where it shows up in the results. For the LS algorithms agreement with the truth is best for the near offsets. Less offset produces less effect of a possible lateral velocity gradient, but limiting offset limits precision (predicted error) and may increase susceptibility to other possible error sources in nature not modeled in the simulator. Notice finally that the Helmert algorithm with a lateral velocity model provides negligible discrepancies with the truth over all offset ranges. These Helmert results compare favorably with predicted coordinate errors implied by the quality of the simulated picks.

Table 2: Real-Data Results minus Acoustics

Algorithm	Offset Range	Mean ΔX	SD ΔX	Mean ΔY	SD ΔY
LS (no gradients)	0-900 ms	5.18 m	3.59 m	-2.56 m	3.90 m
LS (no gradients)	0-470 ms	1.55 m	1.49 m	-0.87 m	2.68 m
LS (no gradients)	470-900 ms	7.05 m	2.88 m	-3.05 m	3.65 m
LS (vertical gradient only)	0-900 ms	5.69 m	2.39 m	-2.50 m	3.09 m
LS (vertical gradient only)	0-470 ms	1.52 m	1.61 m	-0.84 m	2.43 m
LS (vertical gradient only)	470-900 ms	7.01 m	2.88 m	-2.95 m	3.62 m
Helmert (both gradients)	0-900 ms	1.78 m	2.58 m	-1.20 m	2.25 m
Helmert (both gradients)	0-470 ms	1.15 m	1.68 m	-1.46 m	2.23 m
Helmert (both gradients)	470-900 ms	2.23 m	2.82 m	-1.14 m	2.62 m
Helmert (both gradients)	fars - nears	1.08 m	2.26 m	0.32 m	2.03 m

Table 2 gives the results of processing the real data considered in Figures 1 through 6 with the same three algorithms used in Table 1. Notice that offset ranges are in milliseconds, a consequence of a different processing flow in the seismic system since true coordinates are not known. The comparisons in Table 2 are with the results of the acoustic system deployed on the job. Due to its own, very-different error budget, these acoustic results cannot be considered the absolute truth, as is the case with the simulated comparisons. Although the near-offset picks do well in the LS algorithms (which do not model the lateral velocity gradient), the far-offset picks compare poorly with acoustics. By contrast, the Helmert algorithm is effective over all offset ranges. The comparisons of the fars with the nears using the Helmert algorithm (last row) are as good as either against the acoustics. This may imply that the acoustics are biased at the 1-2 meter level. There are many factors that could have caused this. For example, a small but systematic delay between the pinging event and the logging of the GPS coordinates was discovered in the acoustic recording software in this prospect. Although a fix for this delay was made, any uncorrected effect would be in the in-line direction as revealed by the Helmert comparisons of Table 2. Our objective is to find and extirpate every possible source of error in any deployed positioning system. Nevertheless, the comparisons between Helmert first-break processing over the entire offset range and acoustics are at a level acceptable for seismic processing.

Conclusion

This paper has described several sources of systematic error in first-break pick positioning algorithms and detailed methods for dealing with many of them. Simulated and real data examples have been documented to demonstrate that excellent OBC positioning results are achievable with first-break picks in comparison with truth or acoustics if these potential errors are appropriately modeled. When we are confident that near and far offsets produce statistically-equivalent results we have a strategy for dealing with the last, less-common error source, namely, a complex near-surface geology. That strategy is to process as wide an offset range as is consistent with balanced geometry and a decrease in the predicted error of the coordinates. By processing over, through, under and around anomalies (when, in fact, they exist) we stand the best chance of "averaging out" their potential effect on our final coordinates.

References:

Cross, P.A., *Advanced least squares for position fixing*, North East London Polytechnic, 1983

Appendix A: Modeling a Lateral Velocity Gradient

After the global polynomial regression described above models a vertical velocity gradient and converts pick times (perhaps in milliseconds) into pick time distances (perhaps in meters) for processing into coordinates by least-squares methods, a lateral velocity gradient may remain. Such a lateral gradient may be approximated as a quadratic surface, that is, a two-dimensional, second-order polynomial in spatial coordinates over the prospect area, as follows:

Eq A1 $$pvt = b_0 + b_1 \cdot x + b_2 \cdot y + b_3 \cdot x^2 + b_4 \cdot y^2 + b_5 \cdot x \cdot y$$

where *pvt* is "point velocity trend", *x* and *y* are the grid coordinates of a particular point in the prospect, and b_0 through b_5 are the polynomial coefficients common to the entire swath of OBC data being processed. Solved in a simultaneous, least-squares algorithm, these coefficients, which are a function of all picks among the many source and detector points in the swath, provide a uniformly-varying gradient that adequately models reality.

How is such a least-squares algorithm constructed? First, we need to write an observation equation that links our known with our unknowns. We know the pick time distance, which is a produced by our global regression. We need to refine our coordinates (*x*, *y*) and our polynomial coefficients (b_0 through b_5). If the possible existence of a lateral velocity gradient were not an issue, our observation equation would be that normally used in positioning algorithms, namely:

Eq A2 $$ptd = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

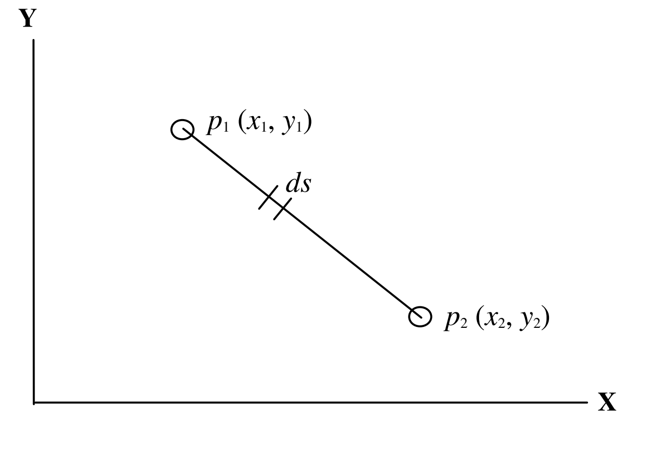
where *ptd* is the pick time distance and the coordinates of p_1 (x_1, y_1) or p_2 (x_2, y_2) or both are unknown. For simplicity of exposition we assume that only the detector coordinates are unknown, but the method can be generalized. To compute the coefficients of the design matrix, discussed later, the non-linear observation equation for *ptd* must be linearized by expanding it about a seed position in a Taylor's series and preserving only the first-order terms, a common procedure in geodetic adjustments. Effectively, this requires partial differentiation with respect to the unknown parameters, in this case each of the detector coordinates. Coordinate seed values are substituted for the parameters to determine the coefficients of the design matrix for the first iteration. Parameter values are updated in subsequent iterations until convergence is achieved and non-linearities are removed.

Since we allow the possibility of a lateral velocity gradient, our observation equation now becomes:

Eq A3 $$ptd = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \, / \, V,$$

where *V* is the average relative velocity trend over the travel path of the seismic energy, a number near unity. Since we have removed the effects of global static delays and the vertical velocity gradient in the global regression stage, we assume the travel path is effectively horizontal.

Figure A1



Now, we must formulate an equation for *V* in terms of the coordinates of p_1 and p_2 and the lateral polynomial coefficients, some or all of which must be refined in the successive iterations of the algorithm. In Figure A1, let p_1 be the shot and p_2 be the detector. The coordinate grid in *x* and *y* is shown and *ds* is a differentially small element of the straight-line path between shot and detector. Of course, given the particular characteristics of the lateral velocity gradient, the straight-line path may not be the fastest path followed by the energy, but it's a useful approximation that simplifies the derivations to follow. Using a common procedure in calculus, we redefine *ds* as follows:

Eq A4 $$ds = \sqrt{(dx)^2 + (dy)^2} = \sqrt{1 + (dy/dx)^2} \cdot dx = \sqrt{1 + (\frac{y_2 - y_1}{x_2 - x_1})^2} \cdot dx$$

The last step is possible because *dy*/*dx* is the slope of the line from p_1 to p_2. To further simplify the mathematics, we now redefine the lateral velocity as its inverse, namely:

Eq A5 $$pvt = 1/(a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot x^2 + a_4 \cdot y^2 + a_5 \cdot x \cdot y),$$

that is, we actually model the inverse of a lateral velocity gradient. One is easily mapped into the other. Except for mathematical convenience, it is immaterial which approach we take.

Now, we apply some basic physics to solve for the relative time it takes for energy to travel from p_1 to p_2, namely a reformulation of distance equals velocity times time:

Eq A6 $$t = s / v,$$

where t is time, s is distance and v is velocity. Expressed differentially, this is:

Eq A7 $$dt = \frac{1}{v} \cdot ds = \frac{1}{pvt} \cdot ds,$$

where pvt is understood to vary over the path from p_1 to p_2 according to Eq A5 above. Total time is found by integrating this function from p_1 to p_2. This integration is simplified by another substitution for the slope-intercept equation of the line between p_1 and p_2, over which we later integrate, as follows:

Eq A8 $$y = (\frac{y_2 - y_1}{x_2 - x_1}) \cdot x + \frac{x_2 \cdot y_1 - x_1 \cdot y_2}{x_2 - x_1}$$

I call the right-hand expression of Eq A8 *yterm*. It is a function of x and the coordinates of the endpoints of the line, which we can treat as knowns in this context. Making all the above substitutions, our integral in the variable x is now:

Eq A9 $$T = \int_{x_1}^{x_2} (a_0 + a_1 \cdot x + a_2 \cdot yterm + a_3 \cdot x^2 + a_4 \cdot yterm^2 + a_5 \cdot x \cdot yterm) \cdot \sqrt{1 + (\frac{y_2 - y_1}{x_2 - x_1})^2} \cdot dx,$$

where T represents the total relative time it takes energy to travel from p_1 to p_2. This integral is easily solved analytically using commercial mathematics software. The answer is:

Eq A10 $$T = S \cdot mterm,$$

where S is the Pythagorean distance between p_1 and p_2 and *mterm* is a "magic" term that comes up again in these computations, namely:

$$\frac{6a_0+3a_1(x_1+x_2)+3a_2(y_1+y_2)+2a_3(x_1^2+x_1x_2+x_2^2)+2a_4(y_1^2+y_1y_2+y_2^2)+a_5(x_1(2y_1+y_2)+x_2(y_1+2y_2))}{6}$$

Now, since T is total relative time, the average relative velocity V over S is as follows:

Eq A11 $$V = \frac{S}{T} = \frac{S}{S \cdot mterm} = \frac{1}{mterm}$$

Substituting into our observation equation (Eq A3) that contains coefficients of a lateral velocity gradient, we have:

Eq A12 $$ptd = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \cdot mterm$$

In the case of detector positioning, the coefficients of our design matrix are the partials of *ptd* with respect to the two detector coordinates and the six polynomial coefficients. Here a simplifying assumption and commercial mathematics software come to the rescue. The simplifying assumption is that the detector coordinates in *mterm* can be held fixed while differentiating with respect to those coordinates in the other term, the Pythagorean distance, S. Coordinate errors in *mterm* have only a small effect on the velocity gradient and, if ignored, their effect will be refined as the algorithm iterates to convergence. Our eight design matrix coefficients are:

Eq A13 $$\frac{\partial(ptd)}{\partial(x_2)} = (x_2 - x_1) \cdot mterm / S$$

Eq A14 $$\frac{\partial(ptd)}{\partial(y_2)} = (y_2 - y_1) \cdot mterm / S$$

Eq A15 $$\frac{\partial(ptd)}{\partial(a_0)} = S$$

Eq A16 $$\frac{\partial(ptd)}{\partial(a_1)} = (x_2 + x_1) \cdot S / 2$$

Eq A17 $$\frac{\partial(ptd)}{\partial(a_2)} = (y_2 + y_1) \cdot S / 2$$

Eq A18 $$\frac{\partial(ptd)}{\partial(a_3)} = (x_1^2 + x_1 \cdot x_2 + x_2^2) \cdot S / 3$$

Eq A19 $$\frac{\partial(ptd)}{\partial(a_4)} = (y_1^2 + y_1 \cdot y_2 + y_2^2) \cdot S / 3$$

Eq A20 $$\frac{\partial(ptd)}{\partial(a_5)} = (x_1 \cdot (2y_1 + y_2) + x_2 \cdot (y_1 + 2y_2)) \cdot S / 6$$

The equations above are sufficient to solve for a quadratic, lateral velocity gradient simultaneously with all detector coordinates in a network adjustment by anyone skilled in constructing such algorithms. Consider a small swath with 4,000 source locations (shots) and 200 detectors and 20,000 first breaks recorded within acceptable offset ranges, a modest 100 picks per detector. Ignoring for simplicity the necessary complication of observational uncertainties and the "weight" matrix, the unweighted form of the matrix equation is the following:

Eq A21 $$r = H \cdot p + e,$$

where *r* is a 20,000 by 1 column vector of residuals (computed minus observed pick time distances), *p* is a 406 by 1 column vector of unknown parameters (200 detectors times 2 coordinates each plus 6 polynomial coefficients), *e* is a 20,000 by 1 column vector of unknown errors to be minimized by the least-squares constraint and *H* is 20,000 by 406 design matrix of coefficients as computed above (Equations A13 through A20). *H* will be very sparse. Each of the 20,000 rows of length 406 will consist of just eight non-zero elements, one each for the two coordinates of the involved detector in the columns defined by their rows in *p* and six for the polynomial coefficients, always in the same columns. Because the lateral velocity gradient is common to the entire swath, i.e. the polynomial coefficients are represented in every observation equation, it is necessary that all detectors be solved in a simultaneous network adjustment. The unweighted least-squares solution is:

Eq A22 $$p = (H' \cdot H)^{-1} \cdot H' \cdot r \, ,$$

where H' is the transpose of H. Due to the need to remove the non-linearities of the observation equation and the approximations made in deriving the partials for the polynomial coefficients in the design matrix, the algorithm must be iterated to convergence. Due to the size and sparsity of the design matrix, computation will be slow.

In my implementation the lateral velocity model is solved much more efficiently by means of a Helmert-blocked least-squares algorithm. Helmert blocking is a least-squares technique that is well known in the geodetic community (ref. Cross). Generally, Helmert blocking is used to simplify continental geodetic adjustments by defining the common block to consist of the coordinates of geodetic stations along country borders and the several local blocks to consist of the interior geodetic points of the several countries. By blocking the adjustment, computational efficiencies are achieved and the confidentiality of the interior coordinates are preserved by the respective countries. My implementation changes the usual procedure by defining the common block to consist of the coefficients a_0 through a_5, which are, indeed, common to all sources and detectors, and the numerous local blocks to consist of the coordinates of the detectors or sources or both and, optionally, a static offset term for the detectors, if desired. Sparsity is eliminated and computational efficiency is greatly enhanced. The details of the Helmert reformulation of this network least-squares problem are complex and beyond the scope of this paper. The reader is referred to Cross (pages 115-124).

The Helmert solution is also iterative. Observations and drop coordinates are globally regressed to compute pick time distances for each pick time with the global vertical velocity gradient removed. Using nominal source and detectors coordinates and the pick time distances, the lateral velocity model coefficients are computed. Using the lateral velocity model coefficients and some intermediate matrix products, the coordinates of the sources or detectors or both are updated. If the difference between the nominal coordinates and the updated coordinates is less than some defined convergence tolerance, we are done. If not, then updated coordinates replace nominal coordinates and updated lateral velocity coefficients replace nominal coefficients on the second iteration and the process repeats. Iteration continues until the difference between the output coordinates and the input coordinates for a given iteration is less than the defined convergence tolerance.

Figure 1: OBC Swath Subset Analyzed in This Paper

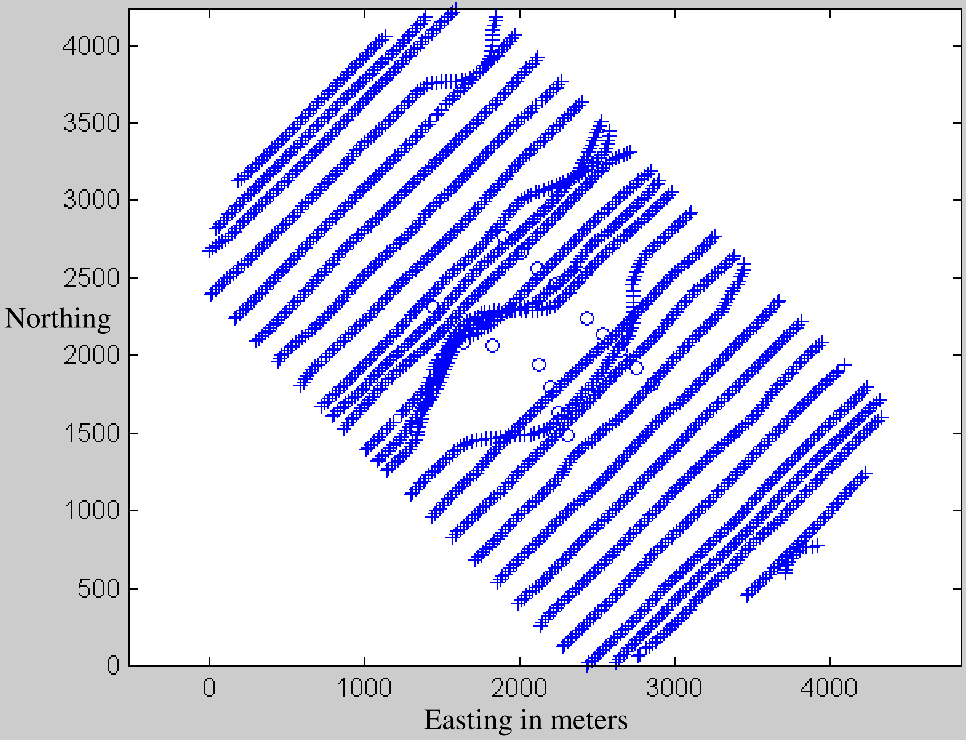


Figure 2: Offset Versus Pick Time Before Repositioning

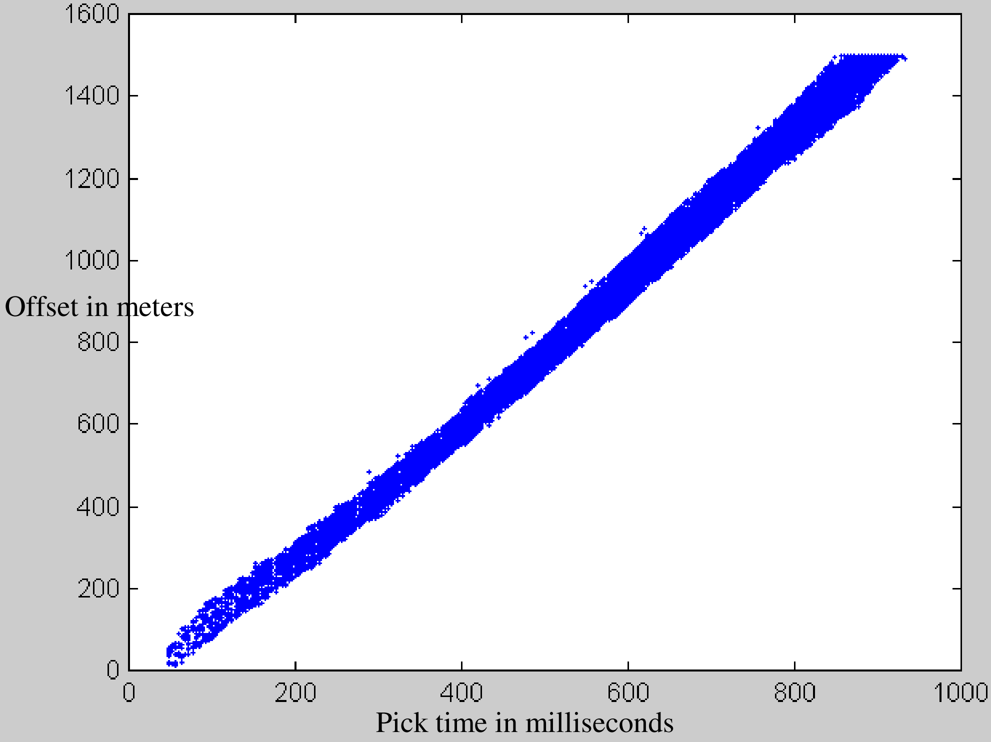


Figure 3: Fourth-Order Transverse Polynomial Fit to the Data

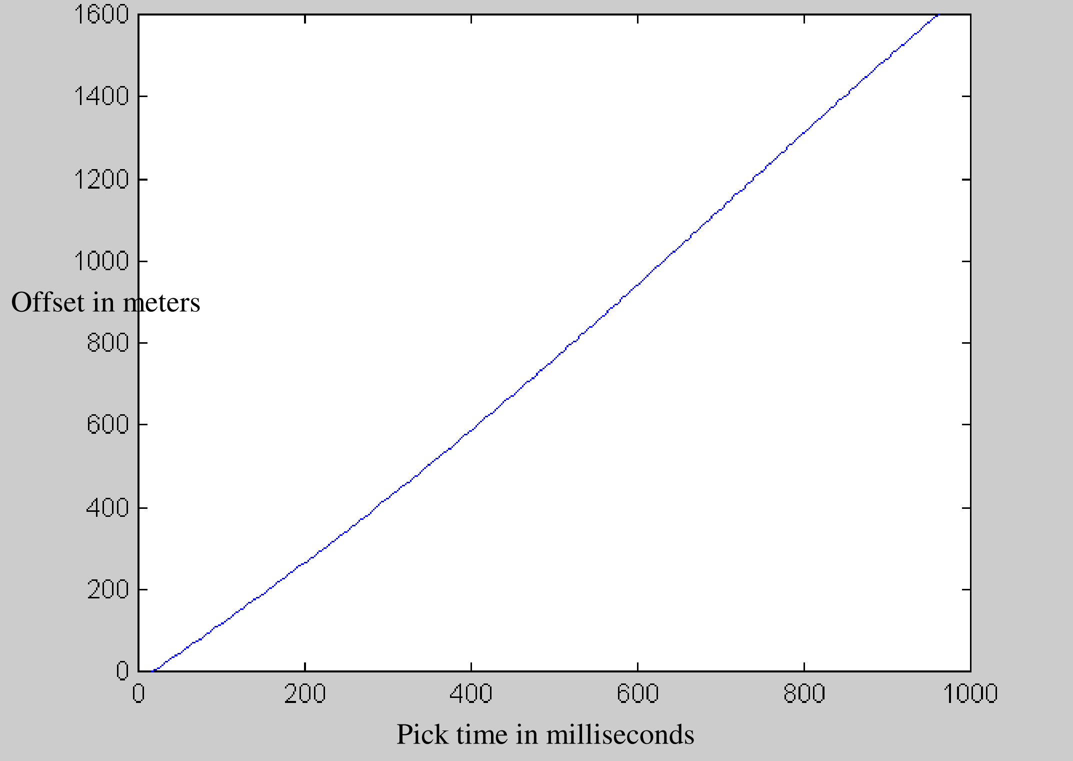


Figure 4: Residuals Versus Pick Time with First-Order Polynomial

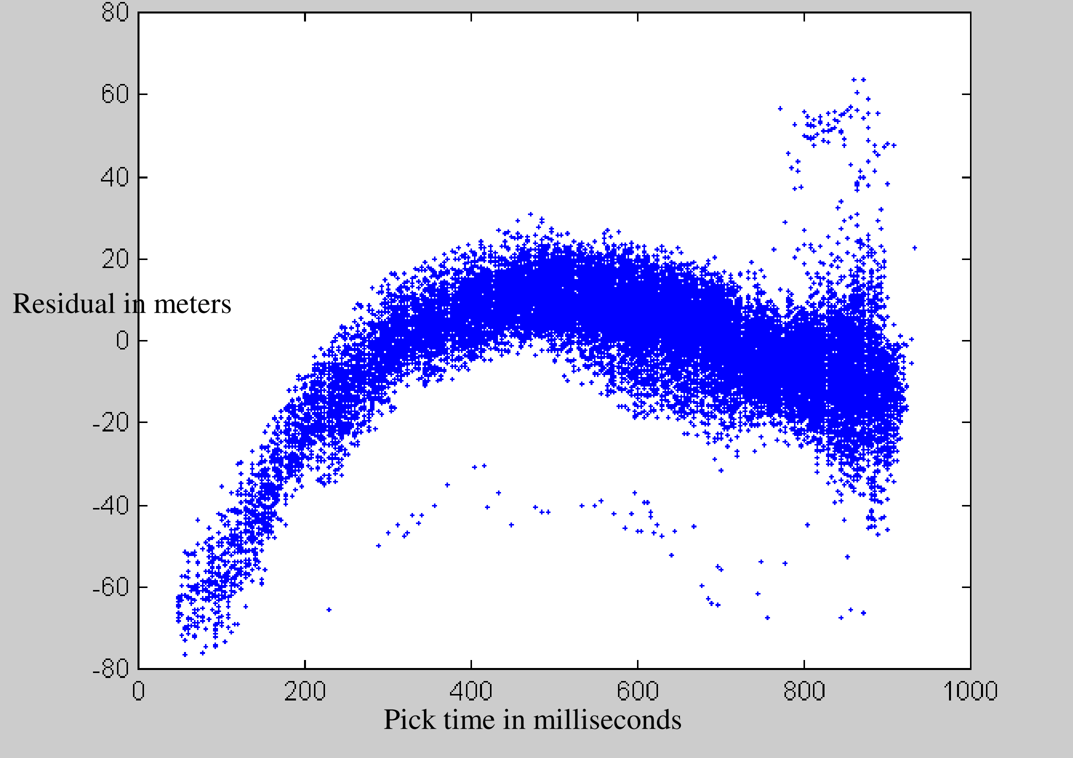


Figure 5: Residuals Versus Pick Time with Fifth-Order

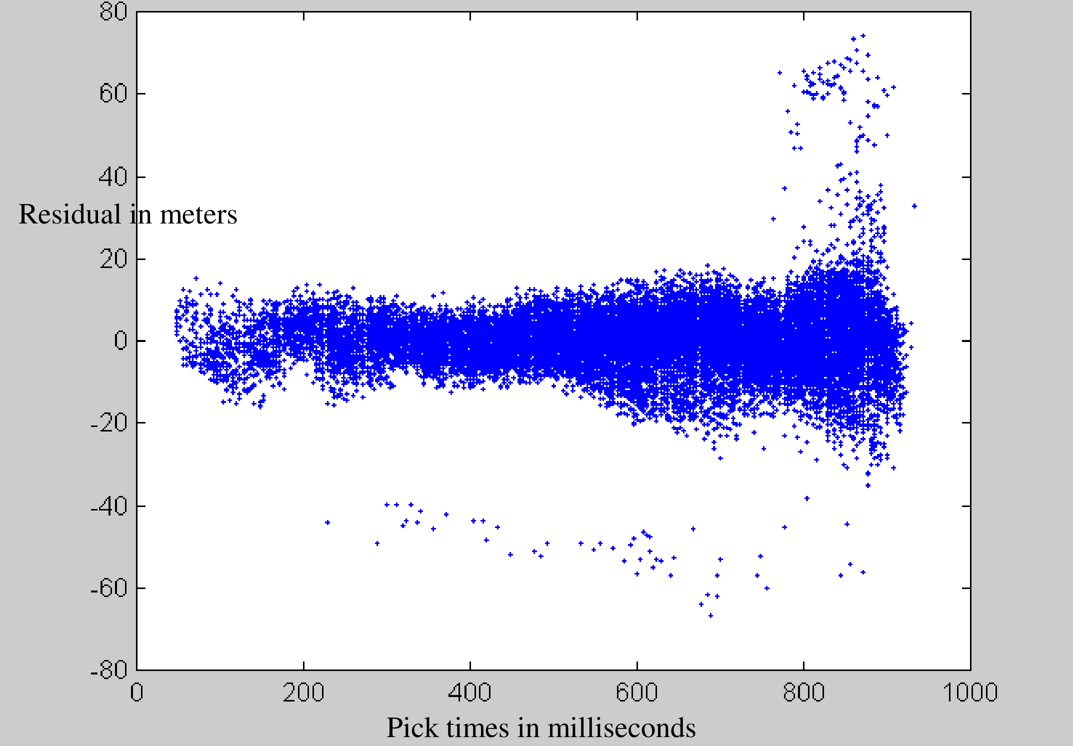


Figure 6: Quadratic Lateral Velocity Surface

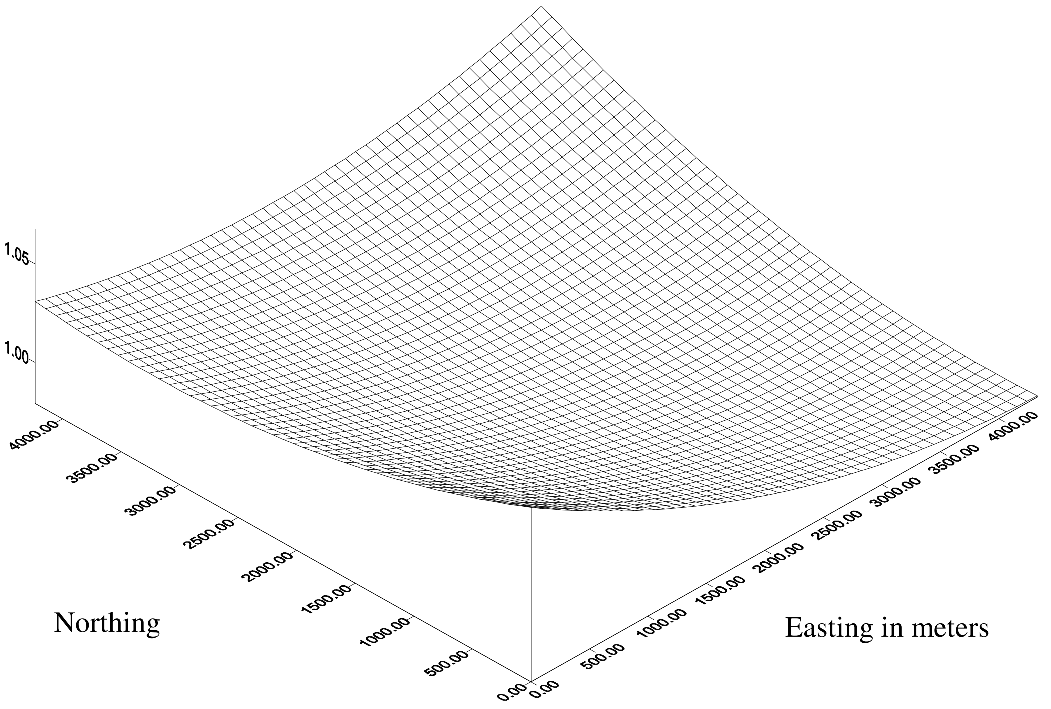


Figure 7: Helmert Algorithm Flow Diagram

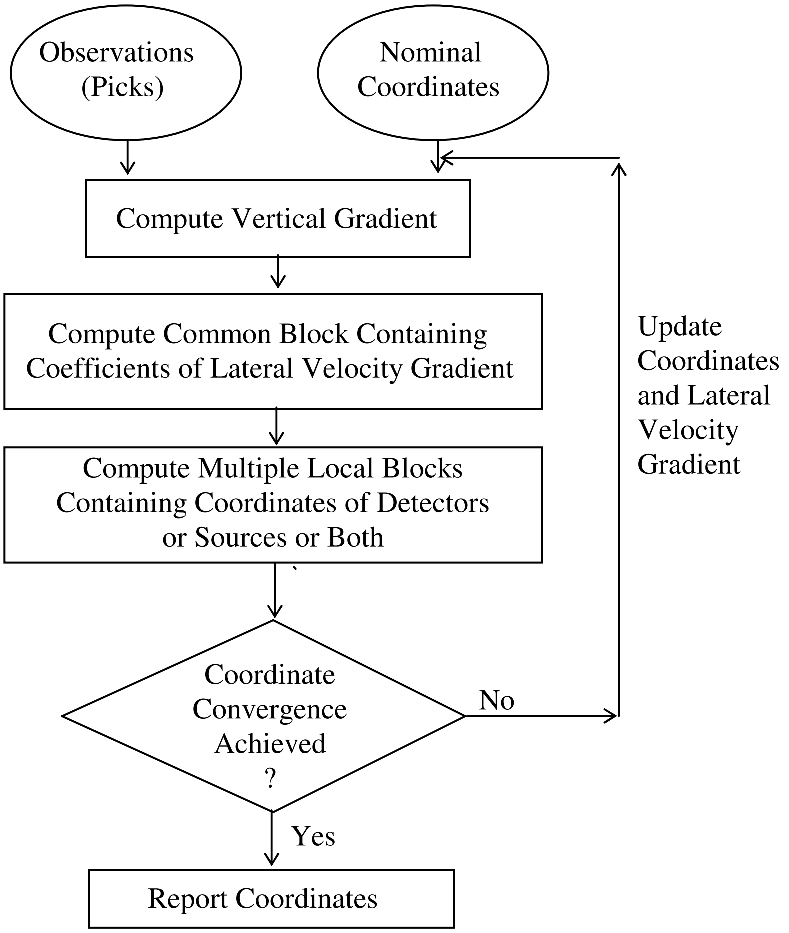
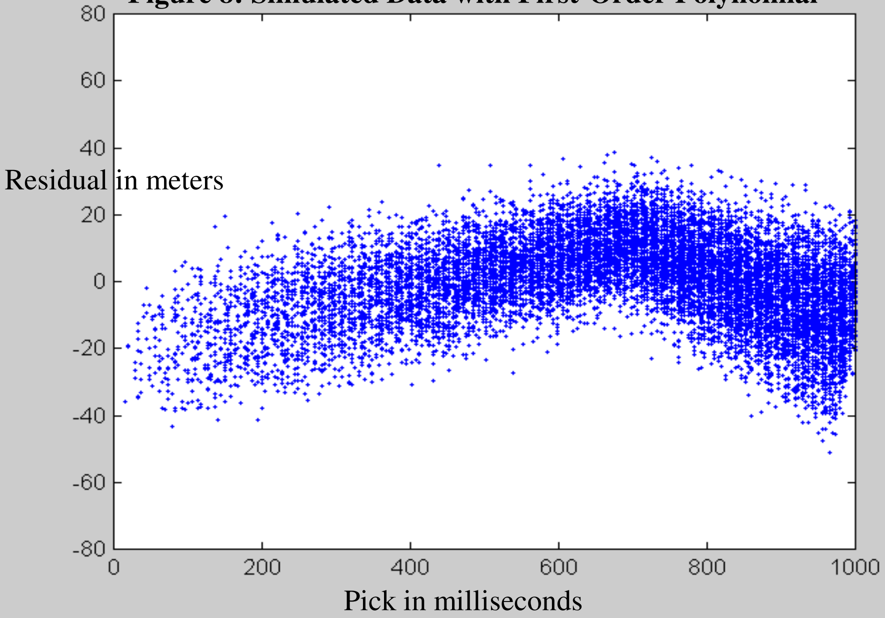

Observations (Picks)
Nominal Coordinates
Compute Vertical Gradient
Compute Common Block Containing Coefficients of Lateral Velocity Gradient
Compute Multiple Local Blocks Containing Coordinates of Detectors or Sources or Both
Coordinate Convergence Achieved ?
No
Update Coordinates and Lateral Velocity Gradient
Yes
Report Coordinates

Figure 8: Simulated Data with First-Order Polynomial


Residual in meters
Pick in milliseconds