Chapter 1

Least Squares

1.1 Basic Statistics

*This section will summarize basic statistics, how to form a hypothesis test, types of error, (t, χ^2, F) Tests, assumptions made, etc.

Summary of Common Statistical Tests

Here I want to add a list of tests and how to do them. So for example, are these two populations the same. Is this distance significant, etc etc.

1.2 Confidence Intervals

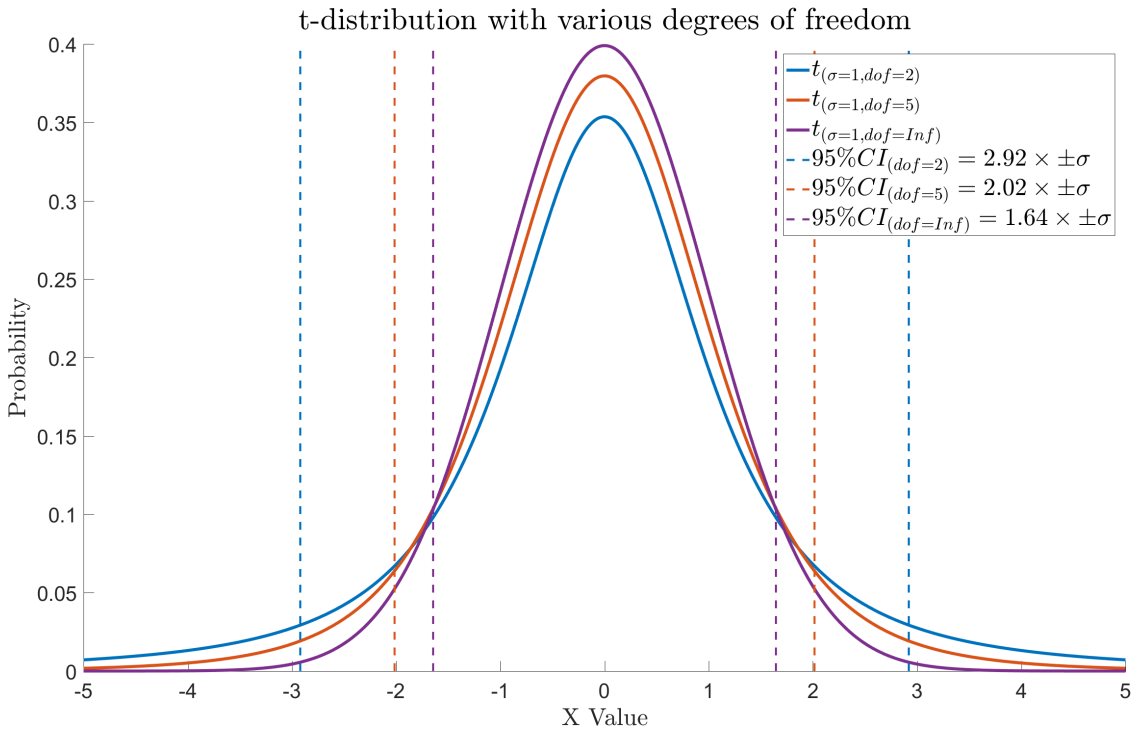
section needs a lot of work

Confidence Intervals are used to convey the uncertainty in a measurement. For example, if the distance from point A to point B is measured infinite times with a mean of $\overline{AB} = 10.00m$ and a standard deviation of $\sigma_{\overline{AB}} = 0.10m$. This measurement could be written as $\overline{AB} = 10.00m \pm 0.10m$. Based on this information, each of the following statements is an accurate confidence interval:

- The true value of the distance $\overline{AB}$ is expected to lie between 9.90 and 10.10 68.2% of the time
- The true value of the distance $\overline{AB}$ is expected to lie between 9.80 and 10.20, 95.4% of the time.
- The true value of the distance $\overline{AB}$ is expected to lie between 9.60 and 10.40, 99.9937% of the time.

*Note: I'm not really sure if confidence intervals are open or closed... and I'm not sure it matters since the odds that value will be a finite number is 0%. In other words, the odds of the true value lying on the edge of a confidence interval is so low that it really shouldn't matter.

The previous example is comparing the mean of an observation, the confidence interval is governed by the t-distribution. The t-distribution is based on the degrees of freedom, which in this case we defined as being infinitely large. When the degrees of freedom are infinite, the t distribution is equivalent to the normal distribution. As the degrees of freedom become smaller, the confidence in the mean and standard deviation get worse, and therefore the confidence interval increases, as shown below.



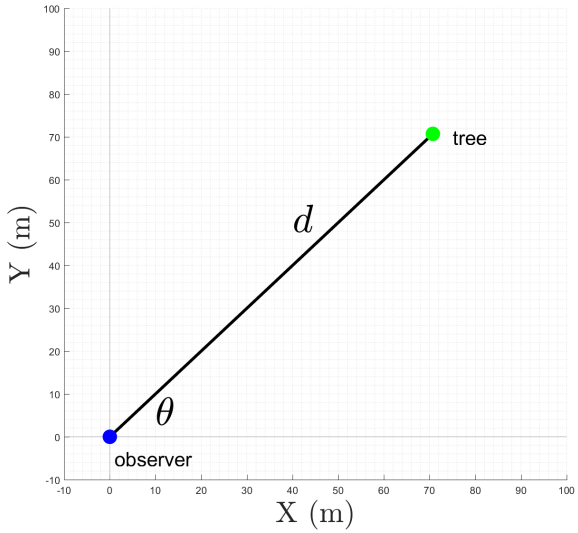
1.2.1 2D Case (Error Rectangle)

1.3 Covariance Matrices

Covariance Matrices (Variance-Covariance Matrices), are used to depict the uncertainty of variables and the correlation between the uncertainty of variables, and are an integral component to least squares. The visualization and calculation of the covariance matrices will be covered later, but this section is to introduce the general concept of a covariance matrix. For example:

Lets say you are standing at point (0,0) in a local coordinate system. Using a compass you determine that a tree is located at about a 45 degree azimuth from true north, and about 100 meters away. Using trigonometry, you can report the coordinate using the equations:

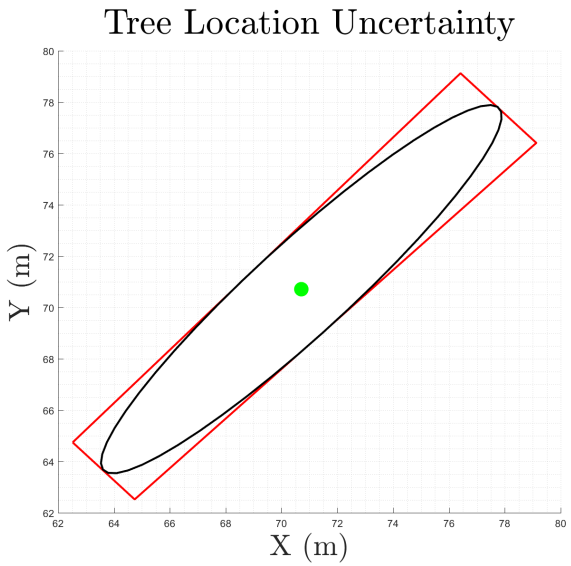
$$x_{tree} = d\cos(az)$$
$$y_{tree} = d\sin(az)$$



If you determined that you had some uncertainty in your measurements:

$$d \pm \sigma_d = 100m \pm 10m$$
$$az \pm \sigma_{az} = 45° \pm 1°$$

The estimate for the position of the tree would look something like the following figure, because there is much more uncertainty in the distance measurement than the angle measurement.



If you were to simply represent the x and y coordinate from this solution using the variance of each component, your uncertainty region would look like a rectangle. But, this would be discarding relevant information from how the coordinate was computed.

Notice that the shape of the most probably region for the tree is an ellipse. As the x estimate is too high, so is the y estimate. As the x estimate is too low, so is the y estimate. The error between the two uncertainties are correlated. This information is what is encompassed in the covariance matrix, where the diagonal elements are the variances of each variable, and the off diagonal elements are the "covariances", which represent the correlation between the errors. In the tree example, the covariances would be positive, as there is a positive correlation between the x and y errors. (As the predicted X increases, the predicted Y increases)

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

The structure of a covariance matrix in a more general example looks like this: Given three variables (x_1, x_2, x_3), V represents a variance, and C represents a covariance.

$$\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1x_2} & \sigma_{x_1x_3} \\ \sigma_{x_1x_2} & \sigma_{x_2}^2 & \sigma_{x_2x_3} \\ \sigma_{x_1x_3} & \sigma_{x_2x_3} & \sigma_{x_3}^2 \end{bmatrix} = \begin{bmatrix} V & C & C \\ C & V & C \\ C & C & V \end{bmatrix}$$

Covariance matrices must have a specific structure. A covariance matrix must:

- Be Symmetrical
- Be Positive Semi-Definite (positive eigen values)
- Have positive variances on the diagonal
- Contain Real numbers (no imaginary)

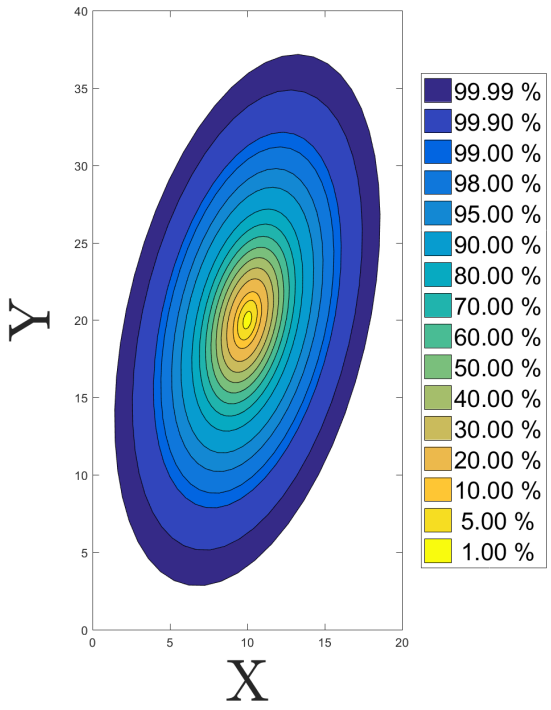
1.4 Error Ellipses and Ellipsoids

Given the point (X, Y):

$$X = 10 \pm 2$$
$$Y = 20 \pm 4$$

Error ellipses(2D) and ellipsoids(3D) are used to depict the region in which a point is expected to lie within a percentage of the time. The percentage of the time the point is expected to lie within the region indicates the confidence. For example, a 1% confidence region is much smaller than a 99% confidence region, as shown in the figure below. In other words, based on the uncertainty in the measurement, the true value will only lie within the 1% error ellipse 1% of the time. The center of an error ellipse is the *most probable value*.

$$\Sigma = \begin{bmatrix} \sigma_X^2 & \sigma_{XY} \\ \sigma_{XY} & \sigma_Y^2 \end{bmatrix} = \begin{bmatrix} 4 & 3 \\ 3 & 10 \end{bmatrix}$$



Note that the *Standard Error Rectangle* is a rectangle centered on the most probable value, with a width of $2\sigma_X$ and a height of $2\sigma_Y$. It is another way to depict the expected region in which a value lies.

1.4.1 Calculation of Semi-Major and Semi-Minor Axes

need to add this section

1.5 Error Propagation (GLOPOV/SLOPOV)

Unknown values (Z) are often calculated from measurements (x), which contain uncertainty (Σ_x).
For example:

$$\begin{aligned} Z_1 &= F_1(x_1, x_2, \ldots, x_n) \\ Z_2 &= F_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ Z_m &= F_m(x_1, x_2, \ldots, x_n) \end{aligned}$$

With the Covariance Matrix:

$$\begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1x_2} & \ldots & \sigma_{x_1x_n} \\ \sigma_{x_1x_2} & \sigma_{x_2}^2 & \ldots & \sigma_{x_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_1x_n} & \sigma_{x_2x_n} & \ldots & \sigma_{x_n}^2 \end{bmatrix}$$

The propagation of uncertainty is governed by the *general law of propagation of variances* (GLOPOV).

1.5.1 GLOPOV

The equation for GLOPOV is given as:

$$\Sigma_{ZZ} = J_{yx}\Sigma J_{yx}^T$$

$$\Sigma_{ZZ} = \begin{bmatrix} \frac{\partial Z_1}{\partial x_1} & \frac{\partial Z_1}{\partial x_2} & \ldots & \frac{\partial Z_1}{\partial x_n} \\ \frac{\partial Z_2}{\partial x_1} & \frac{\partial Z_2}{\partial x_2} & \ldots & \frac{\partial Z_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Z_m}{\partial x_1} & \frac{\partial Z_m}{\partial x_2} & \ldots & \frac{\partial Z_m}{\partial x_n} \end{bmatrix} \times \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1x_2} & \ldots & \sigma_{x_1x_n} \\ \sigma_{x_1x_2} & \sigma_{x_2}^2 & \ldots & \sigma_{x_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_1x_n} & \sigma_{x_2x_n} & \ldots & \sigma_{x_n}^2 \end{bmatrix} \times \begin{bmatrix} \frac{\partial Z_1}{\partial x_1} & \frac{\partial Z_2}{\partial x_1} & \ldots & \frac{\partial Z_m}{\partial x_1} \\ \frac{\partial Z_1}{\partial x_2} & \frac{\partial Z_2}{\partial x_2} & \ldots & \frac{\partial Z_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial Z_1}{\partial x_n} & \frac{\partial Z_2}{\partial x_n} & \ldots & \frac{\partial Z_m}{\partial x_n} \end{bmatrix}$$

1.5.2 SLOPOV

With **no covariances**, and only **one function Z**, GLOPOV can be simplified by the *special law of propagation of variances* SLOPOV:

$$\Sigma = \begin{bmatrix} \sigma_{x_1}^2 & 0 & \ldots & 0 \\ 0 & \sigma_{x_2}^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{x_n}^2 \end{bmatrix}$$

$$\sigma_{Z_1} = \sqrt{(\frac{\partial Z_1}{\partial x_1}\sigma_{x_1})^2 + (\frac{\partial Z_1}{\partial x_2}\sigma_{x_2})^2 + \cdots + (\frac{\partial Z_1}{\partial x_n}\sigma_{x_n})^2}$$

1.5.3 GLOPOV Example

The slope and y-intercept of a linear line are calculated using least squares. The values are:

$$m = 1.25 \qquad b = 0.3 \qquad \Sigma_x = \begin{bmatrix} 0.2 & -1 \\ -1 & 10 \end{bmatrix}$$

Calculate the expected value and uncertainty of points:

$$x_1 = 3.0 \pm 0 \qquad x_2 = 5.0 \pm 0.2$$

Using GLOPOV:

$$\begin{aligned} Z_1 : &\qquad y = mx_1 + b = (1.25)(3.0) + (0.3) \\ Z_2 : &\qquad y = mx_2 + b = (1.25)(5.0) + (0.3) \end{aligned}$$

$$\Sigma = \begin{bmatrix} \sigma_m^2 & \sigma_{mb} & 0 & 0 \\ \sigma_{mb} & \sigma_b^2 & 0 & 0 \\ 0 & 0 & \sigma_{x_1}^2 & 0 \\ 0 & 0 & 0 & \sigma_{x_2}^2 \end{bmatrix} = \begin{bmatrix} 0.2 & -1 & 0 & 0 \\ -1 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}$$

$$A = \begin{bmatrix} \frac{\partial Z_1}{\partial m} & \frac{\partial Z_1}{\partial b} & \frac{\partial Z_1}{\partial x_1} & \frac{\partial Z_1}{\partial x_2} \\ \frac{\partial Z_2}{\partial m} & \frac{\partial Z_2}{\partial b} & \frac{\partial Z_2}{\partial x_1} & \frac{\partial Z_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} x_1 & 1 & m & 0 \\ x_2 & 1 & 0 & m \end{bmatrix} = \begin{bmatrix} 3.0 & 1 & 1.25 & 0 \\ 5.0 & 1 & 0 & 1.25 \end{bmatrix}$$

$$\Sigma_{ZZ} = A\Sigma A^T = \begin{bmatrix} 3.0 & 1 & 1.25 & 0 \\ 5.0 & 1 & 0 & 1.25 \end{bmatrix} \begin{bmatrix} 0.2 & -1 & 0 & 0 \\ -1 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix} \begin{bmatrix} 3.0 & 5.0 \\ 1 & 1 \\ 1.25 & 0 \\ 0 & 1.25 \end{bmatrix} = \begin{bmatrix} 5.80 & 5.00 \\ 5.00 & 5.31 \end{bmatrix}$$

1.5.4 SLOPOV Example

$$\begin{aligned} L &= 10.1cm \pm 0.25cm \\ W &= 4.7cm \pm 0.03cm \\ H &= 6.3cm \pm 0.10cm \end{aligned}$$

Calculate the volume of the fish tank using the equation:

$$\begin{aligned} V &= LWH \\ V &= (10.1cm)(4.7cm)(6.3cm) \\ V &= 299.06cm^3 \end{aligned}$$

Solve Estimated σ using SLOPOV:

$$\begin{aligned} \sigma_V &= \sqrt{(\frac{\partial V}{\partial L}\sigma_L)^2 + (\frac{\partial V}{\partial W}\sigma_W)^2 + (\frac{\partial V}{\partial H}\sigma_H)^2} \\ \sigma_V &= \sqrt{(WH(0.25cm))^2 + (LH(0.03cm))^2 + (LW(0.10cm))^2} \\ \sigma_V &= \sqrt{((4.7cm)(6.3cm)(0.25cm))^2 + ((10.1cm)(6.3cm)(0.03cm))^2 + ((10.1cm)(4.7cm)(0.10cm))^2} \\ \sigma_V &= 9.00cm^3 \end{aligned}$$

Answer: $V = 299.06cm^3 \pm 9.00cm^3$

1.6 Derivations of Least Squares

Least Squares regression solves an over-constrained set of equations as a minimization of the residuals squared(v_i). There are a few different least squares regression methods, which will be described in this section.

$$min \sum_{i=1}^{n} v_i^2$$

1.6.1 Ordinary and Weighted Least Squares

Ghalani provides a good basic example explanation of a simple ordinary least squares problem (Section 11.6.1 Ghalani). Consider the 3 equations provided below.

Equations

The system is overconstrained and there is no solution that satisfies each equation.

$$\begin{aligned} x + y &= 3.0 \\ 2x - y &= 1.5 \\ x - y &= 0.2 \end{aligned}$$

Observation Equations

In order to solve the solution, we add a *residual* for each equation. This residual captures any error in the measurements. These equations are called the observation equations, and are a foundation of least squares.

$$\begin{aligned} x + y &= 3.0 + v_1 \\ 2x - y &= 1.5 + v_2 \\ x - y &= 0.2 + v_3 \end{aligned}$$

Residual Equations

The observation equations can be rewritten as residual equations, shown below.

$$\begin{aligned} v_1 &= x + y - 3.0 \\ v_2 &= 2x - y - 1.5 \\ v_3 &= x - y - 0.2 \end{aligned}$$

The goal of least squares is to determine a solution which minimizes the sum of the square of the residuals... or to get the **least squares solution!**

$$min(\sum_{i=1}^{n} v_i^2)$$

Sum of Squares Equation

$$f(x, y) = (x + y - 3.0)^2 + (2x - y - 1.5)^2 + (x - y - 0.2)^2$$

The minimum of the sum of squares is where the derivative in the x and y dimension is equal to 0. Therefore we derive a normal equation as the partial derivative with respect to each unknown variable, in this case (x,y).

Normal Equation Derivation

$$\frac{\partial f(x,y)}{\partial x} = 2(x+y-3.0) + 2(2x-y-1.5)(2) + 2(x-y-0.2)$$
$$\frac{\partial f(x,y)}{\partial y} = 2(x+y-3.0) + 2(2x-y-1.5)(-1) + 2(x-y-0.2)(-1)$$

Normal Equations

$$6x - 2y - 6.2 = 0$$
$$-2x + 3y - 1.3 = 0$$

Solving for the normal equations yields the 'most probable solution'.

Solving Normal Equations Using Matrices

$$\begin{bmatrix} 6 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6.2 \\ 1.3 \end{bmatrix}$$
$$\begin{bmatrix} x \\ y \end{bmatrix} = inv\left(\begin{bmatrix} 6 & -2 \\ -2 & 3 \end{bmatrix} \right) \begin{bmatrix} 6.2 \\ 1.3 \end{bmatrix}$$
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.514 \\ 1.443 \end{bmatrix}$$

Derive Normal Equations Using Matrices

If we write our initial observation equations as matrices...

$$AX = L + V$$
$$A = \begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 1 & -1 \end{bmatrix} \qquad X = \begin{bmatrix} x \\ y \end{bmatrix} \qquad L = \begin{bmatrix} 3.0 \\ 1.5 \\ 0.2 \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

Pre-multiplying both sides of the equation by A^T, we get the same Normal Equations as above! (V goes away because the normal equation can be solved without residuals). $A^T A$ is sometimes rewritten as N, or the Normal Matrix.

$$AX = L + V$$
$$A^T AX = A^T L \rightarrow \text{ sometimes written as } NX = A^T L$$
$$\begin{bmatrix} 1 & 2 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 3.0 \\ 1.5 \\ 0.2 \end{bmatrix}$$
$$\begin{bmatrix} 6 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 6.2 \\ 1.3 \end{bmatrix}$$

Notice that this is the same as the normal equations derived above.

Derive Matrix Least Squares Solution (Ordinary Least Squares)

Putting the pieces together, we can now solve for the least squares solution using matrices where X represents the most probable values which are the result of a minimization of the squared residuals of each observation.

$$AX = L + V$$
$$A^T AX = A^T L$$
$$X = inv(A^T A) A^T L$$

Add a Weight Term (Stochastic Model)

The weight matrix, W, is used to weight the importance of each observation equation. This is used when you have higher confidence in certain observations, and want those observations to more significantly influence your result. While these weights can be related to anything, it is most common in geomatics that the W matrix is equal to the covariance matrix for all of the observation equations. Normally each observation is assumed to be statistically independent, which would yield a diagonal matrix with no covariances. This is often refereed to as Weighted Least Squares(WLS). However, in an example using GPS surveying, 3 observation equations (X,Y,Z) with covariances are created for each measurement. When Covariances are present, this is often referred to as General Least Squares(GLS). The equations to solve the system are exactly the same. Note: Ghilani Textbook does not use the WLS-GLS nomenclature, as he describes GLS the same way others describe TLS online, so I've chosen to follow the "internet" method for this document.

An example weight matrix for 4 observations equations (a,b,c,d) is shown below for WLS and GLS.

$$\text{Weighted Least Squares(WLS) } W = \Sigma^{-1} = \begin{bmatrix} \sigma_{aa}^2 & 0 & 0 & 0 \\ 0 & \sigma_{bb}^2 & 0 & 0 \\ 0 & 0 & \sigma_{cc}^2 & 0 \\ 0 & 0 & 0 & \sigma_{dd}^2 \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\sigma_{aa}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma_{bb}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\sigma_{cc}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma_{dd}^2} \end{bmatrix}$$

$$\text{General Least Squares(GLS) } W = \Sigma^{-1} = \begin{bmatrix} \sigma_{aa}^2 & \sigma_{ab}^2 & \sigma_{ac}^2 & \sigma_{ad}^2 \\ \sigma_{ab}^2 & \sigma_{bb}^2 & \sigma_{bc}^2 & \sigma_{bd}^2 \\ \sigma_{ac}^2 & \sigma_{bc}^2 & \sigma_{cc}^2 & \sigma_{cd}^2 \\ \sigma_{ad}^2 & \sigma_{bd}^2 & \sigma_{cd}^2 & \sigma_{dd}^2 \end{bmatrix}^{-1}$$

Weighted Linear Least Squares and General Least Squares Equation

Incorporating the Weight matrix results in the following, where A^TWA is the Normal Matrix.

$$\begin{aligned} WAX &= WL + WV \\ A^TWAX &= A^TWL \\ X &= inv(A^TWA)A^TWL \end{aligned}$$

1.6.2 Nonlinear Least Squares

Each of the previous examples only considered linear observation equations, so the values in the A matrix were constants. When this is not the case, an iterative nonlinear least squares approach must be used. Instead of writing the observation equation as $AX = L + V$ as in the linear case, we write the observation equations as $JX = K + V$. Where J is a matrix of partial derivatives of each observation with respect to each unknown. This method is based on taylor series expansion, and it requires a guess at the initial X (X_0).

The X, in this case, is sometimes written as a ΔX, as it represents a delta from the initial guess. The delta is applied to the initial guess, and it a new delta is solved... which is then applied. This iterative approach is performed until it stabilizes on a solution. The exit criteria for the loop is normally based on when the reference variance S_0^2 increases. This either means, your solution is diverging to a poor solution, or the deltas have become so small, that they are "bouncing" up and down about the solution. Other exit criteria are a finite number of iterations, or a small absolute value of the delta values for each unknown. Incorporating a maximum loop iterations is a good idea so that the while loop does not spin off into oblivion.

Weighted Nonlinear Least Squares

Incorporating the Weight Matrix results in the following nonlinear least squares solution.

$$\begin{aligned} WJ\Delta X &= WK + WV \\ J^T WJ\Delta X &= J^T WK \\ \Delta X &= inv(J^T WJ)J^T WK \end{aligned}$$

1.6.3 Total Least Squares (TLS)

*This section is derived from the "General Least Squares" section of Ghilani... From reading online, I believe this is actually called TLS by most other communities.

It is important to recognize that the weight in all of the previous examples is associated with each observation equation, and not each individual variable. For example, when solving a coordinate transformation, it is impossible to have error in both the reference and transformed coordinates. Total Least Squares is required to allow residuals in each of the variables.

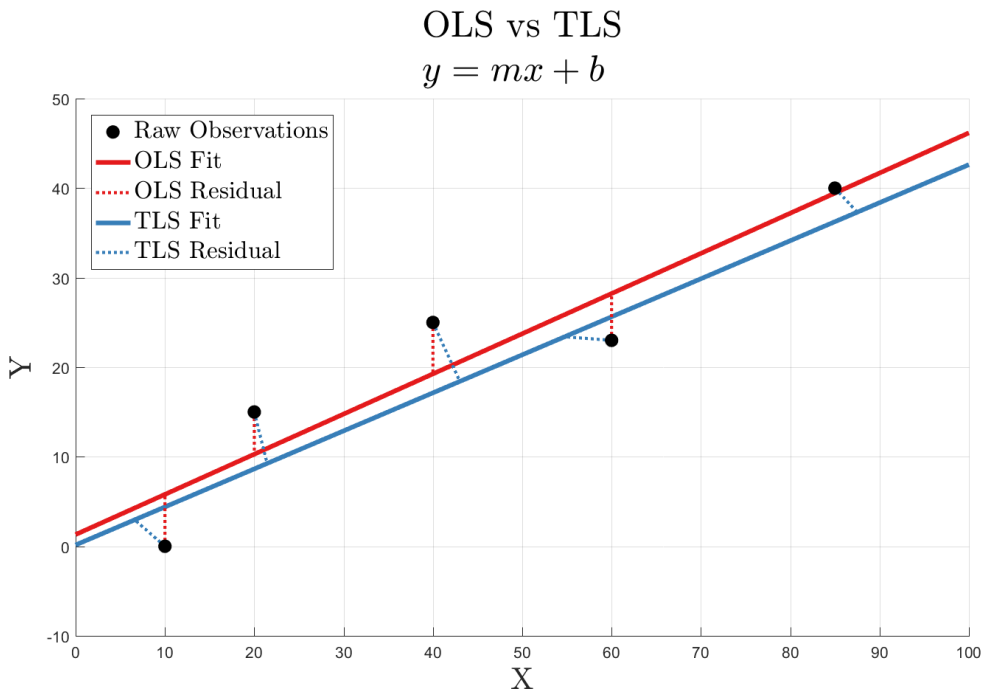
Consider the least squares fit of a line with the observation equation to fit a linear line through a set of data points. Notice that the residual is associated with the y coordinate.

$$mx + b = y + v$$

This is almost always the methodology performed when you do a linear regression with most software packages. If, however, you want to have error in both the x and y coordinate, you must use TLS. The observation equation would then be:

$$m(x + v_x) + b = (y + v_y)$$

A visual plot showing how the fit from TLS is different is shown below:



Total Least Squares Solution

Define your observation equations such that all of the observations and residuals are on the left side of the equation.

$$\begin{aligned} mx + b &= y \\ m(x + v_x) + b &= (y + v_y) \\ m(x + v_x) + b - (y + v_y) &= 0 \end{aligned}$$

Now we want to linearize the equation with respect to the unknowns and the observations.

$$F : m(x + v_x) + b - (y + v_y) = 0$$

$$\begin{aligned} \frac{\partial F}{\partial x} v_x + \frac{\partial F}{\partial y} v_y + \frac{\partial F}{\partial m} d_m + \frac{\partial F}{\partial b} d_b &= 0 - (m_0 x + b_0 - y) \\ m v_x - v_y + x d_m + d_b &= 0 - (mx + b - y) \end{aligned}$$

Now if we had 4 points (A,B,C,D), we could write the following equations.

$$\begin{aligned} m v_{x_A} - v_{y_A} + x_A d_m + d_b &= 0 - (m x_A + b - y_A) \\ m v_{x_B} - v_{y_B} + x_B d_m + d_b &= 0 - (m x_B + b - y_B) \\ m v_{x_C} - v_{y_C} + x_C d_m + d_b &= 0 - (m x_C + b - y_C) \\ m v_{x_D} - v_{y_D} + x_D d_m + d_b &= 0 - (m x_D + b - y_D) \end{aligned}$$

Which can be rewritten in Matrix form:

$$BV + JX = K$$

Where:

$$B = \begin{bmatrix} m & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m & -1 \end{bmatrix} \qquad V = \begin{bmatrix} v_{x_A} \\ v_{y_A} \\ v_{x_B} \\ v_{y_B} \\ v_{x_C} \\ v_{y_C} \\ v_{x_D} \\ v_{x_D} \end{bmatrix}$$

$$J = \begin{bmatrix} x_A & 1 \\ x_A & 1 \\ x_A & 1 \\ x_A & 1 \end{bmatrix} \qquad X = \begin{bmatrix} dm \\ db \end{bmatrix} \qquad K = \begin{bmatrix} -(m x_A + b - y_A) \\ -(m x_B + b - y_B) \\ -(m x_C + b - y_C) \\ -(m x_D + b - y_D) \end{bmatrix}$$

With an optional Covariance Matrix for all of the observations equal to:

$$\Sigma = \begin{bmatrix} \sigma^2_{x_A x_A} & \sigma^2_{x_A y_A} & \sigma^2_{x_A x_B} & \sigma^2_{x_A y_B} & \sigma^2_{x_A x_C} & \sigma^2_{x_A y_C} & \sigma^2_{x_A x_D} & \sigma^2_{x_A y_D} \\ \sigma^2_{x_A y_A} & \sigma^2_{y_A y_A} & \sigma^2_{y_A x_B} & \sigma^2_{y_A y_B} & \sigma^2_{y_A x_C} & \sigma^2_{y_A y_C} & \sigma^2_{y_A x_D} & \sigma^2_{y_A y_D} \\ \sigma^2_{x_A x_B} & \sigma^2_{y_A x_B} & \sigma^2_{x_B x_B} & \sigma^2_{x_B y_B} & \sigma^2_{x_B x_C} & \sigma^2_{x_B y_C} & \sigma^2_{x_B x_D} & \sigma^2_{x_B y_D} \\ \sigma^2_{x_A y_B} & \sigma^2_{y_A y_B} & \sigma^2_{x_B y_B} & \sigma^2_{y_B y_B} & \sigma^2_{y_B x_C} & \sigma^2_{y_B y_C} & \sigma^2_{y_B x_D} & \sigma^2_{y_B y_D} \\ \sigma^2_{x_A x_C} & \sigma^2_{y_A x_C} & \sigma^2_{x_B x_C} & \sigma^2_{y_B x_C} & \sigma^2_{x_C x_C} & \sigma^2_{x_C y_C} & \sigma^2_{x_C x_D} & \sigma^2_{x_C y_D} \\ \sigma^2_{x_A y_C} & \sigma^2_{y_A y_C} & \sigma^2_{x_B y_C} & \sigma^2_{y_B y_C} & \sigma^2_{x_C y_C} & \sigma^2_{y_C y_C} & \sigma^2_{y_C x_D} & \sigma^2_{y_C y_D} \\ \sigma^2_{x_A x_D} & \sigma^2_{y_A x_D} & \sigma^2_{x_B x_D} & \sigma^2_{y_B x_D} & \sigma^2_{x_C x_D} & \sigma^2_{y_C x_D} & \sigma^2_{x_D x_D} & \sigma^2_{x_D y_D} \\ \sigma^2_{x_A y_D} & \sigma^2_{y_A y_D} & \sigma^2_{x_B y_D} & \sigma^2_{y_B y_D} & \sigma^2_{x_C y_D} & \sigma^2_{y_C y_D} & \sigma^2_{x_D y_D} & \sigma^2_{y_D y_D} \end{bmatrix}$$

Or with no covariances:

$$\Sigma = \begin{bmatrix} \sigma^2_{x_A} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{y_A} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{x_B} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{y_B} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{x_C} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{y_C} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{x_D} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{y_D} \end{bmatrix}$$

This system of equations is solved by finding an "equivalent solution" by using an "equivalent weight matrix".

$$W_e = inv(B\Sigma B^T)$$

The "Equivalent Weight Matrix System" is:

$$J^T W_e J \Delta X = J^T W_e K$$
$$\Delta X = inv(J^T W_e J) J^T W_e K$$

This is a nonlinear problem which must be solved iteratively. The solution is just like nonlinear least squares, except the Weight term is now also be calculated each iteration.

To calculate the residual for each observation, user the following equations, where $\hat{X}_i$ is the current loop guess at the unknowns:

$$\text{Equivalent Residuals} = V_{eq} = J\hat{X}_i - K$$
$$\text{Observation Residuals} = V = \Sigma B^T W_{eq} V_{eq}$$

1.7 Robust and Resistant Regression

Robust Regression attempts to minimize the weight of outliers. So instead of minimizing the square of the residuals (L2 Norm Regression aka Least Squares), one example would be to minimize the absolute value of the residuals (L1 Norm Regression aka Least Absolute Difference LAD). This minimizes the impact of outliers, as their residuals are no longer squared. A more sophisticated example would use "**Iterative Reweighted Least Squares**", where each point receives a new weight based on an equation for the residuals. **Resistant Least Squares** attempts to remove the outlier observations altogether in an iterative manner.

These methods are useful for datasets where there are likely outliers or blunders.
*I'm not going to go into too much detail on this right now, but this link seems like a good resource for more a more detailed description: https://onlinecourses.science.psu.edu/stat501/node/353

1.8 Variables and Definitions

1.8.1 Calculating Least Squares Parameters

When solving the least squares problem, you can also report other information about the solution. The definition of each of these terms is provided here, and the equation to calculate each term for each least squares method is provided in the summaries.

Residuals (V)

The residuals are easy to compute for ordinary least squares. It is simply the predicted value minus the observed value.

For Ordinary Least Squares:

$$\text{Residuals} = V = AX - L$$

Unit Variance (S_0^2) (mse in matlab)

The unit variance is also called "variance of an observation of unit weight" or the "variance of unit weight." Assuming all of the variances used in the initial weight matrix are correct(accurate stochastic model), the a priori value for the unit variance is 1. With a correct stochastic model, the computer reference variance of the solution after a least squares adjustment should also be equal to 1. If it is too high, it indicates that the initial stochastic model was overly optimistic and that the covariance matrix of the observations needed to be scaled up to characterize the observed residuals. If, the reference variance is too low, the initial stochastic model was too conservative, and the covariance matrix needed to be scaled down to characterize the observed residuals. In the case where the computed reference variance is too low, the test result is often ignored since correcting it will not change the adjusted coordinates significantly. This is basically saying you measured your points more accurately than you thought you did. If the computed reference variance is too high, this indicates that there may either be blunders in the data or you simply you didn't measure your points as accurately as you thought you did. To test if the computed reference variance is statistically different than the a priori reference variance (1), use the goodness of fit test.

For ordinary least squares is:

$$\text{Reference Variance} = S_0^2 = \frac{V'V}{dof}$$

χ^2 Goodness of Fit Test (is S_0^2 statistically equal to 1?)

A two tailed χ^2 Goodness of Fit Test is used to determine if there is an issue with your least squares adjustment at an α significance level. α values are normally low (eg. 0.01, 0.05).

$$\begin{aligned} \text{Null Hypothesis } H_0 &: S_0^2 = 1 \\ \text{Alternative Hypothesis } H_a &: S_0^2 \neq 1 \\ \text{Significance } &: \alpha \end{aligned}$$

Test Statistic:

$$\chi^2 = \frac{vS_0^2}{\sigma^2} = \frac{dof \times S_0^2}{1} = dof \times S_0^2$$

Rejection Region:

$$\begin{aligned} \chi^2 &> \chi^2_{(\alpha/2,dof)} \\ \chi^2 &< \chi^2_{(1-\alpha/2,dof)} \end{aligned}$$

If either of the rejection region tests fail at the alpha level, then the null hypothesis is rejected and we can say that the computed reference variance is not equal to the a priori reference variance of 1 at an α significance

level. **More importantly**, and somewhat counter-intuitively, is that if both rejection region tests pass, we say that the computed reference variance is not significantly different than the a priori reference variance at an *alpha* significance level, so **we set the computed reference variance equal to 1!** What we are saying here is that the computed reference variance is just an estimate, and it should be equal to 1. Since that estimate is not statistically different than 1 (which would indicate a blunder or poor stochastic model), we assume that there are no blunders and our stochastic model is correct. If there are no blunders and our stochastic model is correct, the computed reference variance is equal to 1.

Cofactor Matrix (Q)

The Cofactor Matrix is the inverse of the Normal Matrix. When the reference variance is equal to 1, it is the same as the covariance matrix.

need more info here

Covariance Matrix or Unknowns (Σ_x)

The Covariance Matrix (also Variance-Covariance) describes the variance and covariances between the computed unknown parameters. The variance is the diagonal element, and the off diagonal elements represent how the computed unknown parameters are related. The standard deviation of the unknowns is just the square root of the variances along the diagonal.

Covariance Matrix of Observations (Σ_l)

The covariance matrix describes the variance and covariance of each of your observation equations. So for example, if you have observation equations for distances between a few points, and solve for the coordinates of the unknown points. You can then use this to determine the uncertainty in the distances. This is basically using Σ_x and GLOPOV.

Model Skill (R^2, sometimes $\hat{S}$)

The Skill of the model, or R^2, is the amount of variance explained by the model. A R^2 equal to 1 indicates the model perfectly fits the data. A R^2 equal to 0 indicates that the model does not explain any of the variance of the data. **Do not use R^2 for nonlinear least squares**, as the value is not valid.

Extra Sum of Squares Test (Large N approximation)

The model skill will always increase as the number of unknowns increases, so the significant of adding another unknown to a model can be tested with a 'extra sum of squares test.' This test requires that a large number of N observations. The equation below shows the hypothesis test that is performed. to compare if the extra parameter added in model B (with M_2 parameters) is significant compared to model A(with M_1 parameters) at the α confidence level. If the test is true, the extra parameter is not significant.

$$\frac{R_B^2 - R_A^2}{1 - R_B^2} \leq \frac{1}{N}\chi^2_{(\alpha, M_2 - M_1)}$$

RMSE (Root Mean Square Error)

RMSE is another metric to summarize the residuals of the model. A value of 0 indicates a perfect fit. Higher values indicate higher residuals and a less accurate model.

1.9 Ordinary Least Squares (OLS)

1.9.1 Theory

Ordinary Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation.

$$min \sum_{i}^{n} v_i^2$$

1.9.2 Assumptions

- No Outliers/Blunders OLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-contrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v $\rightarrow$ error only in y dimension)

1.9.3 Equations

$$AX = L + V$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \qquad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \qquad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

$$\text{Unknowns} = \hat{X} = inv(A^T A)A^T L$$

$$\text{Residuals} = V = AX - L$$

$$\text{Reference Variance} = S_0^2 = \frac{V'V}{dof}$$

$$\text{Cofactor Matrix} = Q_{xx} = inv(A^T A)$$

$$\text{Covariance Matrix of Unkowns} = \Sigma_{xx} = S_0^2 Q_{xx}$$

$$\text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} = A\Sigma_{xx}A^T$$

$$\text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})}$$

$$\text{Predicted L} = \hat{L} = AX$$

$$\text{R}^2 \text{ (model skill)} = \frac{var(\hat{L})}{var(L)}$$

$$\text{RMSE} = \sqrt{\frac{V^T V}{m}}$$

1.9.4 Sample Problem

Given the points:

$$x = [0, 1, 2, 3, 4] \qquad y = [5, 1, 7, 13, 24]$$

Fit a parabola given the observation equation:

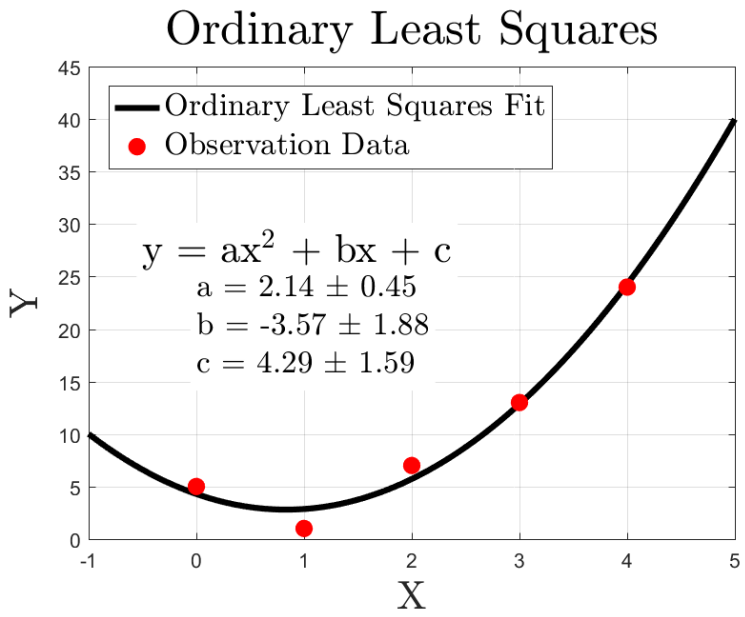
$$y = ax^2 + bx + c$$

Note that the observation equation is linear. The x^2 term is a constant once observation values are substituted.

$$A = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \\ x_5^2 & x_5 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 4 & 2 & 1 \\ 9 & 3 & 1 \\ 16 & 4 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad L = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 7 \\ 13 \\ 24 \end{bmatrix}$$

Use the Equations and solve:

$n = 3$	$m = 5$	$dof = 2$
$\hat{X} = \begin{bmatrix} 2.14 \\ -3.57 \\ 4.29 \end{bmatrix}$	$V = \begin{bmatrix} -0.71 \\ 1.86 \\ -1.29 \\ -0.14 \\ 0.29 \end{bmatrix}$	$S_0^2 = 2.86$
$\Sigma_{xx} = \begin{bmatrix} 0.20 & -0.82 & 0.41 \\ -0.82 & 3.55 & -2.20 \\ 0.41 & -2.20 & 2.53 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.45 \\ 1.88 \\ 1.59 \end{bmatrix}$	$\Sigma_{\hat{l}\hat{l}} = \begin{bmatrix} 2.53 & 0.73 & -0.24 & -0.41 & 0.24 \\ 0.73 & 1.06 & 0.98 & 0.49 & -0.41 \\ -0.24 & 0.98 & 1.39 & 0.98 & -0.24 \\ -0.41 & 0.49 & 0.98 & 1.06 & 0.73 \\ 0.24 & -0.41 & -0.24 & 0.73 & 2.53 \end{bmatrix}$
$Q_{xx} = \begin{bmatrix} 0.07 & -0.29 & 0.14 \\ -0.29 & 1.24 & -0.77 \\ 0.14 & -0.77 & 0.89 \end{bmatrix}$	$\hat{L} = \begin{bmatrix} 4.29 \\ 2.86 \\ 5.71 \\ 12.86 \\ 24.29 \end{bmatrix}$	$R^2 = 0.9821 \qquad RMSE = 1.07$



1.9.5 Example Matlab Code

Algorithm 1.1: exampleOLS.m

```
%% Example Ordinary Least Squares Script
%% observations
x = [0 1 2 3 4];
y = [5 1 7 13 24];

%% Define A and L based on observation equation
A = [x(:).^2 x(:) ones(size(x(:)))];
L = y(:);

%% Calculate Ordinary Least Squares
m = numel(L);                   % number of observations
n = size(A,2);                  % number of unknowns
dof = m-n;                      % degrees of freedom
X = (A'*A)\A'*L;                % unknowns(inv(A)*b' with 'A\b' per Matlab)
V = A * X - L;                  % residuals
So2 = V'*V/dof;                 % Reference Variance
Q = inv(A'*A);                  % cofactor
Sx = So2 * Q;                   % covariance of unknowns
Sl = A * Sx * A';               % covariance of observations
stdX = sqrt(diag(Sx));          % std of solved unknowns
Lhat = A * X;                   % predicted L values
r2 = var(Lhat)/var(L);          % R^2 Skill
RMSE = sqrt(V'*V/m);            % RMSE
```

Algorithm 1.2: The Matlab built in function LSCOV generates the same results

```
%% Using Matlab Built in Function LSCOV
[matlab_X, matlab_stdX, matlab_So2, matlab_Sx] = lscov(A,L); %same results
```

1.10 Weighted Least Squares (WLS)

1.10.1 Theory

Weighted Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation, with a weight (W_i) applied to each observation.

$$min \sum_{i}^{n} W_i \times v_i^2$$

*Note that if the scale of the variance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.10.2 Assumptions

- No Outliers/Blunders WLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-contrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v $\rightarrow$ error only in y dimension)
- No covariance between weight of each observation

1.10.3 Equations

$$WAX = WL + WV$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad W = diag\left(\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \text{ or } \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_m^2 \end{bmatrix}^{-1} \right)$$

$$\text{Unknowns} = \hat{X} = inv(A^T W A) A^T W L$$

$$\text{Residuals} = V = AX - L$$

$$\text{Reference Variance} = S_0^2 = \frac{V'WV}{dof}$$

$$\text{Cofactor Matrix} = Q_{xx} = inv(A^T W A)$$

$$\text{Covariance Matrix of Unkowns} = \Sigma_{xx} = S_0^2 \times Q_{xx}$$

$$\text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} = A \Sigma_{xx} A^T$$

$$\text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})}$$

$$\text{Predicted L} = \hat{L} = AX$$

$$\text{R}^2 \text{ (model skill)} = \frac{var(\hat{L})}{var(L)}$$

$$\text{RMSE} = \sqrt{\frac{VV^T}{m}}$$

1.10.4 Sample Problem

Given the points(x, y) and weights($Weights$):

$$x = [0, 1, 2, 3, 4] \qquad y = [5, 1, 7, 13, 24] \qquad Weights = [1, 10, 100, 5, 1]$$

Fit a parabola given the observation equation:

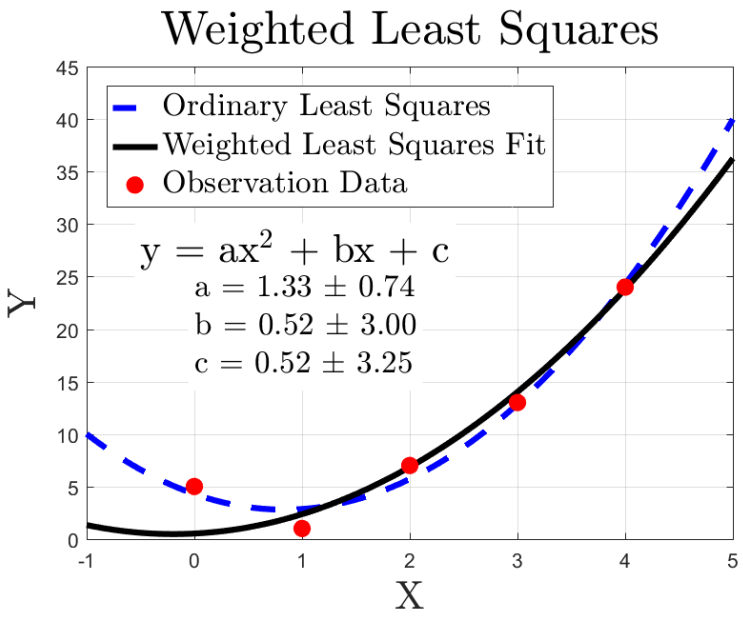
$$y = mx + b$$

Note that the observation equation is linear, and there are no covariances to the weights. These weights have no units, and are just saying hey, I trust the third measurement 10 times more than I trust the second measurement.

$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \qquad L = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 7 \\ 13 \\ 24 \end{bmatrix} \qquad W = \begin{bmatrix} 15 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Use the Equations and solve:

$n = 3$	$m = 5$	$dof = 2$
$\hat{X} = \begin{bmatrix} 1.33 \\ 0.52 \\ 0.52 \end{bmatrix}$	$V = \begin{bmatrix} -4.48 \\ 1.36 \\ -0.14 \\ 1.01 \\ -0.19 \end{bmatrix}$	$S_0^2 = 22.88$
$\Sigma_{xx} = \begin{bmatrix} 0.55 & -2.09 & 1.89 \\ -2.09 & 9.01 & -9.21 \\ 1.89 & -9.21 & 10.59 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.74 \\ 3.00 \\ 3.25 \end{bmatrix}$	$\Sigma_{\hat{l}\hat{l}} = \begin{bmatrix} 10.59 & 3.28 & -0.25 & 0.02 & 4.07 \\ 3.28 & 1.34 & 0.09 & -0.46 & -0.32 \\ -0.25 & 0.09 & 0.22 & 0.14 & -0.15 \\ 0.02 & -0.46 & 0.14 & 1.82 & 4.59 \\ 4.07 & -0.32 & -0.15 & 4.59 & 13.88 \end{bmatrix}$
$Q_{xx} = \begin{bmatrix} 0.02 & -0.09 & 0.08 \\ -0.09 & 0.39 & -0.40 \\ 0.08 & -0.40 & 0.46 \end{bmatrix}$	$\hat{L} = \begin{bmatrix} 0.52 \\ 2.36 \\ 6.86 \\ 14.01 \\ 23.81 \end{bmatrix}$	$R^2 = 1.1366 \qquad RMSE = 2.15$



1.10.5 Example Matlab Code

Algorithm 1.3: exampleWLS.m

```
%% Example Weighted Least Squares Script
%% observations
x = [0 1 2 3 4];
y = [5 1 7 13 24];
weights = [1 10 100 5 1];

%% Define A, L, and W based on observation equation
A = [x(:).^2 x(:) ones(size(x(:)))];
L = y(:);
W = diag(weights);

%% Calculate Weighted Least Squares
m = numel(L);                   % number of observations
n = size(A,2);                  % number of unknowns
dof = m-n;                      % degrees of freedom
X = (A'*W*A)\A'*W*L;            % unknowns(inv(A)*b' with 'A\b' per Matlab)
V = A * X - L;                  % residuals
So2 = V'*W*V/dof;               % Reference Variance
Q = inv(A'*W*A);                % cofactor
Sx = So2 * Q;                   % covariance of unknowns
Sl = A * Sx * A';               % covariance of observations
stdX = sqrt(diag(Sx));          % std of solved unknowns
Lhat = A * X;                   % predicted L values
r2 = var(Lhat)/var(L);          % R^2 Skill
RMSE = sqrt(V'*V/m);            % RMSE
```

Algorithm 1.4: The Matlab built in function LSCOV generates the same results

```
%% Using Matlab Built in Function LSCOV
[mat_X, mat_stdX, mat_So2, mat_Sx] = lscov(A,L,weights); %same results
```

1.11 General Least Squares (GLS)

1.11.1 Theory

General Least Squares solves a linear, overconstrained system of equations by minimizing the squared residuals of each observation equation, with a weight matrix defining the variance AND covariance of each observation equation. Note that the equations are the same as WLS, except the Weight Matrix (W) contains covariances.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.11.2 Assumptions

- No Outliers/Blunders WLS is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is linear (eg. derivative wrt each unknown is not a function of any of the unknowns)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v $\rightarrow$ error only in y dimension)
- Covariance between weight of each observation

1.11.3 Equations

$$WAX = WL + WV$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_m \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad W = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \dots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \dots & \sigma_{mm}^2 \end{bmatrix}^{-1}$$

$$\text{Unknowns} = \hat{X} = inv(A^T W A) A^T W L$$

$$\text{Residuals} = V = AX - L$$

$$\text{Reference Variance} = S_0^2 = \frac{V'WV}{dof}$$

$$\text{Cofactor Matrix} = Q_{xx} = inv(A^T W A)$$

$$\text{Covariance Matrix of Unkowns} = \Sigma_{xx} = S_0^2 \times Q_{xx}$$

$$\text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} = A \Sigma_{xx} A^T$$

$$\text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})}$$

$$\text{Predicted L} = \hat{L} = AX$$

$$\text{R}^2 \text{ (model skill)} = \frac{var(\hat{L})}{var(L)}$$

$$\text{RMSE} = \sqrt{\frac{VV^T}{m}}$$

1.11.4 Sample Problem

Given the control coordinates (x_c, y_c), corresponding measured coordinates (x, y), and estimated covariance matrix of the control coordinates Σ_c:

fix variance covariance EVERYWHERE so only variances are squared... pg88 ghilani

$$x_c = [1, 2, 3] \qquad y_c = [0, 5, 1] \qquad x = [6, 1, 8] \qquad y = [3, 12, 8]$$

$$\Sigma_c = \begin{bmatrix} \sigma^2_{x_{c_1}x_{c_1}} & \sigma^2_{x_{c_1}y_{c_1}} & \sigma^2_{x_{c_1}x_{c_2}} & \sigma^2_{x_{c_1}y_{c_2}} & \sigma^2_{x_{c_1}x_{c_3}} & \sigma^2_{x_{c_1}y_{c_3}} \\ \sigma^2_{x_{c_1}y_{c_1}} & \sigma^2_{y_{c_1}y_{c_1}} & \sigma^2_{y_{c_1}x_{c_2}} & \sigma^2_{y_{c_1}y_{c_2}} & \sigma^2_{y_{c_1}x_{c_3}} & \sigma^2_{y_{c_1}y_{c_3}} \\ \sigma^2_{x_{c_1}x_{c_2}} & \sigma^2_{y_{c_1}x_{c_2}} & \sigma^2_{x_{c_2}x_{c_2}} & \sigma^2_{x_{c_2}y_{c_2}} & \sigma^2_{x_{c_2}x_{c_3}} & \sigma^2_{x_{c_2}y_{c_3}} \\ \sigma^2_{x_{c_1}y_{c_2}} & \sigma^2_{y_{c_1}y_{c_2}} & \sigma^2_{x_{c_2}y_{c_2}} & \sigma^2_{y_{c_2}y_{c_2}} & \sigma^2_{y_{c_2}x_{c_3}} & \sigma^2_{y_{c_2}y_{c_3}} \\ \sigma^2_{x_{c_1}x_{c_3}} & \sigma^2_{y_{c_1}x_{c_3}} & \sigma^2_{x_{c_2}x_{c_3}} & \sigma^2_{y_{c_2}x_{c_3}} & \sigma^2_{x_{c_3}x_{c_3}} & \sigma^2_{x_{c_3}y_{c_3}} \\ \sigma^2_{x_{c_1}y_{c_3}} & \sigma^2_{y_{c_1}y_{c_3}} & \sigma^2_{x_{c_2}y_{c_3}} & \sigma^2_{y_{c_2}y_{c_3}} & \sigma^2_{x_{c_3}y_{c_3}} & \sigma^2_{y_{c_3}y_{c_3}} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.3 & 0 & 0 & 0 & 0 \\ 0.3 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.4 \end{bmatrix}$$

Calculate the 2D Conformal (Four Parameter Similarity Transform) to go from the measured coordinates to the control coordinates. (Note that the block diagonal structure of this covariance matrix describes that the x and y coordinate are correlated to each other for each point, but there is no correlation between points). The 2D Conformal equations are:

equation is wrong!

$$x_c = (S\cos(\theta))x - (S\sin(\theta))y + T_x$$
$$y_c = (S\sin(\theta))x + (S\cos(\theta))y + T_y$$

By substituting:

$$a = S\cos(\theta)$$
$$b = S\sin(\theta)$$

Where:

$$\theta = \tan^{-1}(\frac{b}{a})$$
$$S = \frac{a}{\cos(\theta)}$$

The observation equations become:

$$F: \qquad x_c = ax - by + T_x$$
$$G: \qquad y_c = bx + ay + T_y$$

Note that every $(x_a, y_a) \rightarrow (x, y)$ correspondence produces 2 observation equations.

$$A = \begin{bmatrix} \frac{\partial F_1}{\partial a} & \frac{\partial F_1}{\partial b} & \frac{\partial F_1}{\partial T_x} & \frac{\partial F_1}{\partial T_y} \\ \frac{\partial G_1}{\partial a} & \frac{\partial G_1}{\partial b} & \frac{\partial G_1}{\partial T_x} & \frac{\partial G_1}{\partial T_y} \\ \frac{\partial F_2}{\partial a} & \frac{\partial F_2}{\partial b} & \frac{\partial F_2}{\partial T_x} & \frac{\partial F_2}{\partial T_y} \\ \frac{\partial G_2}{\partial a} & \frac{\partial G_2}{\partial b} & \frac{\partial G_2}{\partial T_x} & \frac{\partial G_2}{\partial T_y} \\ \frac{\partial F_3}{\partial a} & \frac{\partial F_3}{\partial b} & \frac{\partial F_3}{\partial T_x} & \frac{\partial F_3}{\partial T_y} \\ \frac{\partial G_3}{\partial a} & \frac{\partial G_3}{\partial b} & \frac{\partial G_3}{\partial T_x} & \frac{\partial G_3}{\partial T_y} \end{bmatrix} = \begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ x_3 & -y_3 & 1 & 0 \\ y_3 & x_3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & -3 & 1 & 0 \\ 3 & 6 & 0 & 1 \\ 1 & -12 & 1 & 0 \\ 12 & 1 & 0 & 1 \\ 8 & -8 & 1 & 0 \\ 8 & 8 & 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} a \\ b \\ T_x \\ T_y \end{bmatrix}$$

$$L = \begin{bmatrix} F(X_1) \\ G(X_1) \\ F(X_2) \\ G(X_2) \\ F(X_3) \\ G(X_3) \end{bmatrix} = \begin{bmatrix} x_{a_1} \\ y_{a_1} \\ x_{a_2} \\ y_{a_2} \\ x_{a_3} \\ y_{a_3} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 5 \\ 3 \\ 1 \end{bmatrix} \qquad W = inv(\Sigma_c)$$

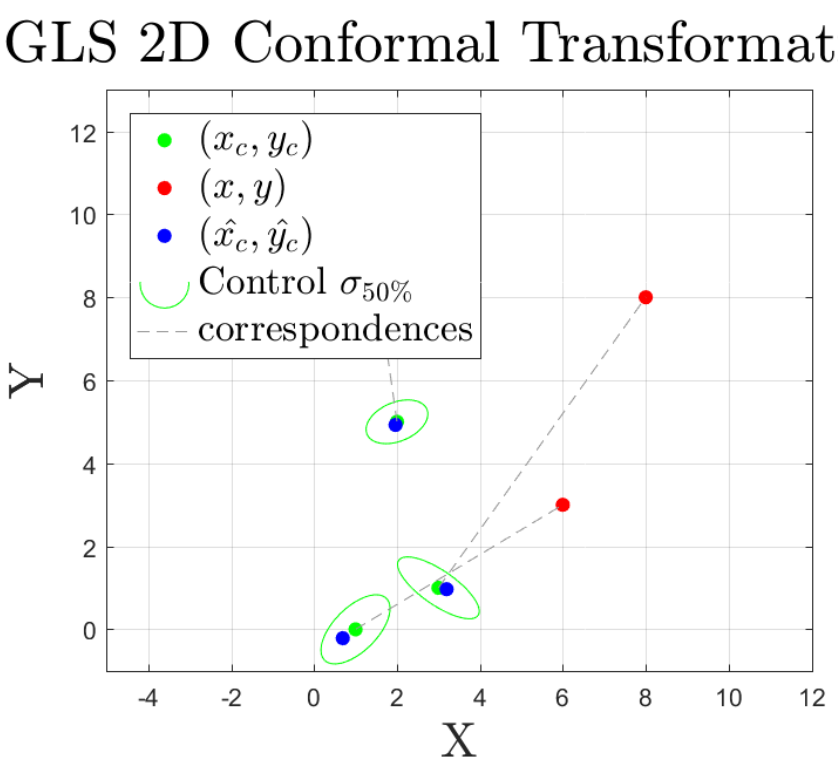
Use the Equations and solve:

$n = 4$	$m = 6$	$dof = 2$
$\hat{X} = \begin{bmatrix} 0.38 \\ -0.35 \\ -2.62 \\ 0.76 \end{bmatrix}$	$V = \begin{bmatrix} -0.31 \\ -0.21 \\ -0.03 \\ -0.07 \\ 0.20 \\ -0.03 \end{bmatrix}$	$S_0^2 = 0.16$
$\Sigma_{xx} = \begin{bmatrix} 0.00 & -0.00 & -0.00 & -0.00 \\ -0.00 & 0.00 & 0.01 & -0.00 \\ -0.00 & 0.01 & 0.08 & 0.01 \\ -0.00 & -0.00 & 0.01 & 0.07 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.02 \\ 0.03 \\ 0.27 \\ 0.27 \end{bmatrix}$	$\Sigma_{\hat{l}\hat{l}} = \begin{bmatrix} 0.03 & 0.00 & 0.00 & -0.01 & 0.01 & 0.01 \\ 0.00 & 0.04 & 0.01 & -0.01 & -0.01 & 0.02 \\ 0.00 & 0.01 & 0.06 & 0.01 & 0.02 & -0.01 \\ -0.01 & -0.01 & 0.01 & 0.03 & 0.01 & -0.00 \\ 0.01 & -0.01 & 0.02 & 0.01 & 0.02 & -0.01 \\ 0.01 & 0.02 & -0.01 & -0.00 & -0.01 & 0.02 \end{bmatrix}$
$Q_{xx} = \begin{bmatrix} 0.00 & -0.00 & -0.02 & -0.03 \\ -0.00 & 0.01 & 0.04 & -0.02 \\ -0.02 & 0.04 & 0.48 & 0.04 \\ -0.03 & -0.02 & 0.04 & 0.47 \end{bmatrix}$	$\hat{L} = \begin{bmatrix} 0.69 \\ -0.21 \\ 1.97 \\ 4.93 \\ 3.20 \\ 0.97 \end{bmatrix}$	$R^2 = 1.1018 \qquad RMSE = 0.18$

Interpret the results:

- θ and S values and their covariances can be calculated using equations: $\theta = \tan^{-1}(\frac{b}{a})$ and $S = \frac{a}{\cos(\theta)}$
- R^2 isn't really a valid metric here, the variance of the coordinate doesn't mean anything useful.
- Error is assumed to only be in the control coordinates
- We can use these results to transform other coordinates with uncertainty using GLOPOV
- The S_0^2 value is not near one, indicating the variance-covariance matrix may have over-estimated the predicted error of the control data. a χ^2 goodness of fit test should be done to confirm this.

legend is messed up


GLS 2D Conformal Transformation
(x_c, y_c)
(x, y)
$(\hat{x}_c, \hat{y}_c)$
Control $\sigma_{50\%}$
correspondences
Y
X
12
10
8
6
4
2
0
-4
-2
0
2
4
6
8
10
12

1.11.5 Example Matlab Code

Algorithm 1.5: exampleGLS.m

```
%% Example General Least Squares Script
%% observations
xc = [1 2 3];
yc = [0 5 1];
x = [6 1 8];
y = [3 12 8];

Sc = [0.5 0.3 0 0 0 0;
      0.3 0.5 0 0 0 0;
      0 0 0.4 0.1 0 0;
      0 0 0.1 0.2 0 0;
      0 0 0 0 0.7 -0.4;
      0 0 0 0 -0.4 0.4]; %variance-covariance of control data

%% Define A, L, and W based on observation equation
A = nan(numel(x)*2,4);
A(1:2:end)=[x(:) -y(:) ones(size(x(:))) zeros(size(y(:)))];
A(2:2:end)=[y(:) x(:) zeros(size(y(:))) ones(size(x(:)))];

L = nan(numel(x)*2,1);
L(1:2:end) = xc;
L(2:2:end) = yc;

W = inv(Sc);

%% Calculate General Least Squares
m = numel(L);                   % number of observations
n = size(A,2);                  % number of unknowns
dof = m-n;                      % degrees of freedom
X = (A'*W*A)\A'*W*L;            % unknowns(inv(A)*b' with 'A\b' per Matlab)
V = A * X - L;                  % residuals
So2 = V'*W*V/dof;               % Reference Variance
Q = inv(A'*W*A);                % cofactor
Sx = So2 * Q;                   % covariance of unknowns
Sl = A * Sx * A';               % covariance of observations
stdX = sqrt(diag(Sx));          % std of solved unknowns
Lhat = A * X;                   % predicted L values
r2 = var(Lhat)/var(L);          % R^2 Skill
RMSE = sqrt(V'*V/m);            % RMSE
```

Algorithm 1.6: The Matlab built in function LSCOV generates the same results

```
%% Using Matlab Built in Function LSCOV
[mat_X, mat_stdX, mat_So2, mat_Sx] = lscov(A,L,Sc); %same results
```

1.12 Nonlinear Least Squares

1.12.1 Theory

Nonlinear Least Squares solves a nonlinear, overconstrained system of equations by minimizing the squared residuals of each observation equation. A Taylor Series expansion is utilized to linearize the equation and iteratively calculate the gradient of the function to determine a local minima. Optionally, a weight matrix(W) may be used. Remove the W term, or replace with an identity matrix if a Weight matrix is not desired. Covariances may be 0s if not known.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.12.2 Assumptions

- No Outliers/Blunders. Nonlinear Least Squares is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System of equations is nonlinear (eg. derivative wrt at least one unknown is a function of one of the other unknowns)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error only in dependent variable (eg. mx+b = y + v $\rightarrow$ error only in y dimension)
- X_0 must be a reasonable guess, otherwise the solution might settle on an incorrect local minima, rather than the global minimum.

1.12.3 Equations

$$AX = L + V$$

$$WJ\Delta X = WK + WV$$

m = number of observations $\qquad$ n = number of unknowns

dof = degrees of freedom (# of redundant observations) $= m - n$

i = loop iteration

Observation Equations $= F_m(x_1, x_2, ..., x_n) = l_m$ (note: AX is obs eqn, L is $[l_1, l_2..., l_m]$)

$$J_i = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix} \quad \Delta X = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \quad K_i = \begin{bmatrix} l_1 - F_1(\hat{X}_i) \\ l_2 - F_2(\hat{X}_i) \\ \vdots \\ l_m - F_m(\hat{X}_i) \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix}$$

$$W = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1m}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2m}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \sigma_{m2}^2 & \cdots & \sigma_{mm}^2 \end{bmatrix}^{-1} \qquad \text{Initial Guess } \hat{X}_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Loop Equations:

Loop until ΔX is small, or more robustly, loop until S_0^2 increases. S_0^2 will increase slightly when you get down to really really small numbers and the cpu starts rounding. Caveat: it will also increase if you have a really bad initial guess, and it starts diverging.

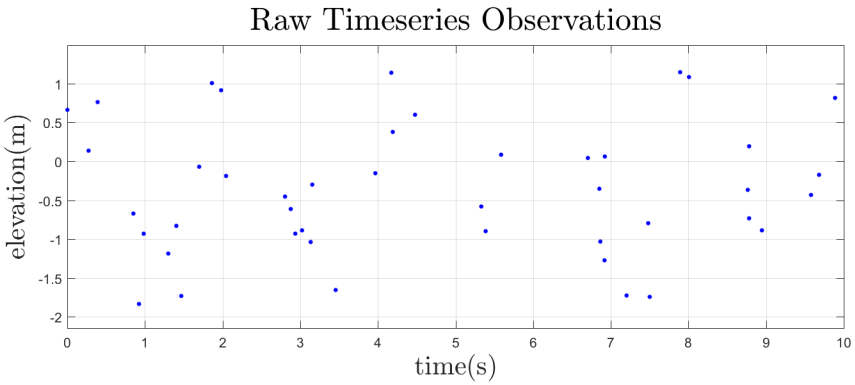
$$\text{Recalculate } K_i = K(\hat{X}_i)$$
$$\text{Recalculate } J_i = J(\hat{X}_i)$$
$$\text{Loop Delta Estimate} = \Delta X = inv(J_i^T W J_i) J_i^T W K_i$$
$$\text{Loop Estimate} = \hat{X}_i = X_{i-1} + \Delta X$$
$$\text{Residuals} = V = K_i$$
$$\text{Reference Variance} = S_0^2 = \frac{V^T W V}{dof}$$

Final Calculations

$$\text{Unknowns} = \hat{X} = \hat{X}_i \text{ (Final Loop Estimate)}$$
$$\text{Cofactor Matrix} = Q_{xx} = inv(J^T W J)$$
$$\text{Covariance Matrix of Unkowns} = \Sigma_{xx} = S_0^2 \times Q_{xx}$$
$$\text{Covariance Matrix of Observations} = \Sigma_{\hat{l}\hat{l}} = J \Sigma_{xx} J^T$$
$$\text{Standard Deviation of Solved Unknowns} = \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})}$$
$$\text{Predicted L} = \hat{L} = AX$$
$$\text{R}^2 \text{ (model skill)} = \text{Not valid for nonlinear least squares}$$
$$\text{RMSE} = \sqrt{\frac{VV^T}{m}}$$

1.12.4 Sample Problem

Given a timeseries dataset (ts.csv) of a harmonic signal with a known period and estimated variance for each measurement, calculate the amplitude and Phase.



The governing equation for the system is:

$$F: \qquad A \times \sin(\frac{2\pi}{T}t + \phi) = y$$

Where:

$$\begin{aligned} \text{Amplitude(m)} &= A \\ \text{Period(s)} &= T = 2 \\ \text{Phase(s)} &= \phi \\ \text{Elevation Measurement(m)} &= y \\ \text{Time Measurement(s)} &= t \end{aligned}$$

Note that this system is Nonlinear due to the Phase term. The Jacobian is solved using the partial derivatives.

$$J_i = \begin{bmatrix} \frac{\partial F_1}{\partial A} & \frac{\partial F_1}{\partial \phi} \\ \frac{\partial F_2}{\partial A} & \frac{\partial F_2}{\partial \phi} \\ \vdots & \vdots \\ \frac{\partial F_m}{\partial A} & \frac{\partial F_m}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \sin(\frac{2\pi}{T}t_1 + \phi_i) & A_i \times \cos(\frac{2\pi}{T}t_1 + \phi_i) \\ \sin(\frac{2\pi}{T}t_2 + \phi_i) & A_i \times \cos(\frac{2\pi}{T}t_2 + \phi_i) \\ \vdots & \vdots \\ \sin(\frac{2\pi}{T}t_m + \phi_i) & A_i \times \cos(\frac{2\pi}{T}t_m + \phi_i) \end{bmatrix}$$

$$\Delta X = \begin{bmatrix} \Delta A \\ \Delta \phi \end{bmatrix} \qquad K_i = \begin{bmatrix} l_1 - F_1(\hat{X}_i) \\ l_2 - F_2(\hat{X}_i) \\ \vdots \\ l_m - F_m(\hat{X}_i) \end{bmatrix} = \begin{bmatrix} y_1 - A_i \times \sin(\frac{2\pi}{T}t_1 + \phi_i) \\ y_2 - A_i \times \sin(\frac{2\pi}{T}t_2 + \phi_i) \\ \vdots \\ y_m - A_i \times \sin(\frac{2\pi}{T}t_3 + \phi_i) \end{bmatrix}$$

$$\text{Initial guess from looking at the plot } = X_0 = \begin{bmatrix} A_0 \\ \phi_0 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1 \end{bmatrix}$$

Solving using Equations:

$n = 2$	$m = 50$	$dof = 48$
$\hat{X} = \begin{bmatrix} 0.38 \\ 0.63 \end{bmatrix}$	$S_0^2 = 0.94$	$Q_{xx} = \begin{bmatrix} 0.02 & -0.00 \\ -0.00 & 0.00 \end{bmatrix}$
$\Sigma_{xx} = \begin{bmatrix} 0.0250 & -0.0137 \\ -0.0137 & 0.1231 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.1353 \\ 0.0499 \end{bmatrix}$	$RMSE = 0.66$

A two tailed χ^2 Goodness of Fit Test is performed at the 0.05 Significance level:

$$\text{Null Hypothesis } H_0 : S_0^2 = 1$$
$$\text{Alternative Hypothesis } H_a : S_0^2 \neq 1$$
$$\text{Significance } : \alpha = 0.05$$

Test Statistic:

$$\chi^2 = \frac{vS_0^2}{\sigma^2} = \frac{dof \times S_0^2}{1} = dof \times S_0^2 = 44.95$$

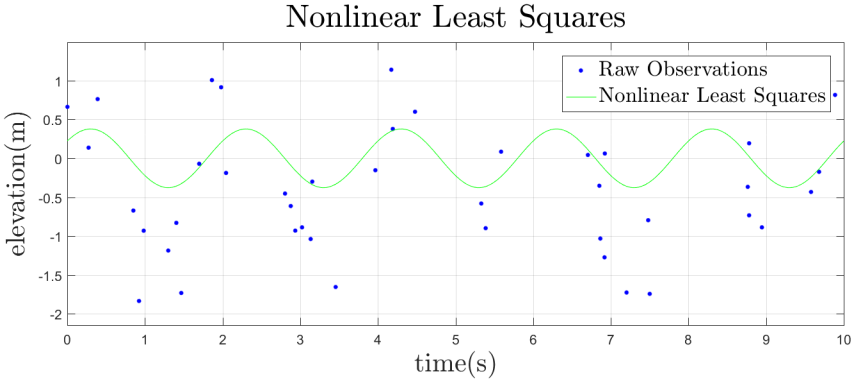
Rejection Region:

$$44.95 = \chi^2 > \chi^2_{(\alpha/2,dof)} = 30.75 \qquad (PASS)$$
$$44.95 = \chi^2 < \chi^2_{(1-\alpha/2,dof)} = 69.02 \qquad (PASS)$$

Results:

The results pass the χ^2 Goodness of Fit Test at the 0.05 significance level, so redo the final calculations with S_0^2 equal to 1.

$S_0^2 = 1.00$	$\Sigma_{xx} = \begin{bmatrix} 0.0196 & -0.0010 \\ -0.0010 & 0.0027 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.1399 \\ 0.0515 \end{bmatrix}$



Note: The previous example could have been calculated using Weighted Least Squares (WLS)
The Observation Equation can be rewritten as:

$$A \times \sin(\frac{2\pi}{T}t) + B \times \cos(\frac{2\pi}{T}t) = y$$

Where A, B, T are unknowns and:

$$Amp = \sqrt{A^2 + B^2}$$
$$\phi = \tan^{-1}(B/A)$$

1.12.5 Example Matlab Code

Algorithm 1.7: exampleNonlinearLS.m

```
%% Example Nonlinear Least Squares Script
%% read data
data=csvread('ts.csv');
t = data(:,1);                  % raw time observations
y = data(:,2);                  % raw elevation observations
stdy = data(:,3);               % reported std for each observation
W = inv(diag(stdy.^2));         % Create a Weight Matrix based on the stdy

%% Solve Least Squares
Xo = [1.5; 1];                   % initial unknowns guess
X = Xo;
So2 = inf; dSo2 = 1; iter = 0;  % initialize while loop

m = numel(t);                       % number of observations
n = numel(X);                   % number of unknowns
dof = m-n;                      % degrees of freedom
while dSo2>=0 && iter<100 %loop until So2 increases or exceed 100 iteration
    J = [sin(2*pi/2.*t + X(2)) X(1)*cos(2*pi/2.*t + X(2))];
    K = y - X(1)*sin(2*pi/2.*t + X(2));
    dX = (J'*W*J)\J'*W*K;       % Loop Delta Estimate
    X = X + dX;                 % Loop Estimate
    V = K;                      % Residuals
    dSo2 = So2 - V'*W*V/dof;    % Change in Reference Variance
    So2 = (V'*W*V)/dof;         % Reference Variance
    iter = iter + 1;
end
Q = inv(J'*W*J);                % cofactor
Sx = So2 * Q;                   % covariance of unknowns
Sl = J * Sx * J';               % covariance of observations
stdX = sqrt(diag(Sx));          % std of solved unknowns
Lhat = J * X;                   % predicted L values
RMSE = sqrt(V'*V/m);            % RMSE

% Perform chi^2 Test
alphaval = 0.05;
chi2 = dof * So2;
chi2low = chi2inv(alphaval/2,dof);
chi2high = chi2inv(1-alphaval/2,dof);
% Recalculate Values with So2 = 1 because CHI2 passed
So2_b = 1;                      % set reference variance equal to 1
Sx_b = So2_b * Q;               % covariance of unknowns     w/ So2 = 1
Sl_b = J * Sx_b * J';           % covariance of observations w/ So2 = 1
stdX_b = sqrt(diag(Sx_b));      % std of solved unknowns     w/ So2 = 1
```

Algorithm 1.8: The Matlab built in function NLINFIT generates the same results

```
%% Using Matlab Built in Function NLINFIT
beta0 = [1.5; 1];
modelfun = @(b,x)(b(1)*sin(2*pi/2*x+b(2)));
options.Display = 'iter';
[mat_X,mat_V,mat_J,mat_Sx,mat_So2,ErrorModelInfo] = ...
```

1.13 Total Least Squares (TLS)

1.13.1 Theory

Total Least Squares solves a nonlinear, overconstrained system of equations by minimizing the squared residuals of each observation. A Taylor Series expansion is utilized to linearize the equation and iteratively calculate the gradient of the function to determine a local minima. A Weight Matrix, with a column/row for each observation rather than observation equation, is used to allow for error in all dimensions. Traditionally, this will be a predicted variance for each observation.

*Note that if the scale of the variance-covariance in the weight matrix is known to be 1, then the computed reference variance should be inspected to ensure it passes the χ^2 goodness of fit test. If it passes the test, the Covariance matrix should NOT be multiplied by the reference variance. See definition of reference variance for the reasoning.

1.13.2 Assumptions

- No Outliers/Blunders. Nonlinear Least Squares is not robust to outliers (consider RANSAC/Robust Weighting if outliers)
- System is over-constrained (eg. Number of Observation Equations > Number of Unknowns)
- Error is in all measurements variable (eg. mx+b = y + v $\rightarrow$ error only in x and y dimension)
- X_0 must be a reasonable guess, otherwise the solution might settle on an incorrect local minima, rather than the global minimum. If a linear problem, one method is to solve the unweighted OLS, then use that to initialize X_0 in TLS.

1.13.3 Equations

$$AX = L \qquad \text{(note: residuals in both X and L)}$$

$$BV + J\Delta X = K$$

$$m = \text{number of observations} \qquad n = \text{number of unknowns}$$

$$p = \text{number of observation equations for each observation}$$

$$q = \text{number of input variables}$$

$$dof = \text{degrees of freedom (\# of redundant observations)} = m - n$$

$$i = \text{loop iteration}$$

$$\text{Observation Equations} = F_m(x_1, x_2, ..., x_n) = l_m \text{ (note: } AX \text{ is obs eqn, L is } [l_1, l_2..., l_m] \text{)}$$

$$b(r) = \begin{bmatrix} \frac{\partial F_1}{\partial v_{1r}} & \frac{\partial F_1}{\partial v_{2r}} & \cdots & \frac{\partial F_1}{\partial v_{pr}} \\ \frac{\partial F_2}{\partial v_{1r}} & \frac{\partial F_2}{\partial v_{2r}} & \cdots & \frac{\partial F_2}{\partial v_{pr}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_k}{\partial v_{1r}} & \frac{\partial F_k}{\partial v_{2r}} & \cdots & \frac{\partial F_k}{\partial v_{pr}} \end{bmatrix} \quad B = \begin{bmatrix} b(1) & 0 & \cdots & 0 \\ 0 & b(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & b(m) \end{bmatrix} \quad V = \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{p1} \\ v_{12} \\ v_{22} \\ \vdots \\ v_{p2} \\ \vdots \\ v_{1m} \\ v_{2m} \\ \vdots \\ v_{pm} \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix} \quad \Delta X = \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{bmatrix} \quad K = \begin{bmatrix} l_1 - F_1(X_i) \\ l_2 - F_2(X_i) \\ \vdots \\ l_m - F_m(X_i) \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1(n\times m)}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2(n\times m)}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{(n\times m)1}^2 & \sigma_{(n\times m)2}^2 & \cdots & \sigma_{(n\times m)(n\times m)}^2 \end{bmatrix} \quad \text{Initial Guess } X_0 = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Loop Equations:

Loop until ΔX is small, or more robustly, loop until S_0^2 increases. S_0^2 will increase slightly when you get down to really really small numbers and the cpu starts rounding. Caveat: it will also increase if you have a really bad initial guess, and it starts diverging.

$$\begin{aligned} \text{Equivalent Weight} &= W_{eq} = inv(B\Sigma B^T) \\ \text{Loop Delta Estimate} &= \Delta X = inv(J^T W_{eq} J) J^T W_{eq} K \\ \text{Loop Estimate} &= \hat{X}_i = X_{i-1} + \Delta X \\ \text{Equivalent Residuals} &= V_{eq} = K_i \\ \text{Reference Variance} &= S_0^2 = \frac{V_{eq}^T W_{eq} V_{eq}}{dof} \end{aligned}$$

make sure V = K in all nonlinear

Final Calculations

$$\begin{aligned} \text{Observation Residuals} &= V = \Sigma B^T W_{eq} V_{eq} \\ \text{Unknowns} &= \hat{X} = \hat{X}_i \text{ (Final Loop Estimate)} \\ \text{Cofactor Matrix} &= Q_{xx} = inv(J^T W_{eq} J) \\ \text{Covariance Matrix of Unkowns} &= \Sigma_{xx} = S_0^2 \times Q_{xx} \\ \text{Covariance Matrix of Observations} &= \Sigma_{\hat{l}\hat{l}} = J\Sigma_{xx} J^T \\ \text{Standard Deviation of Solved Unknowns} &= \sigma_{\hat{X}} = \sqrt{diag(\Sigma_{xx})} \\ \text{RMSE} &= \sqrt{\frac{V_{eq} V_{eq}^T}{m}} \end{aligned}$$

1.13.4 Sample Problem

Given the points and covariance matrix for each observation:

$$x = [10, 20, 60, 40, 85] \qquad y = [0, 15, 23, 25, 40]$$

$$\Sigma = \begin{bmatrix} \sigma^2_{x_1x_1} & \sigma^2_{x_1y_1} & \sigma^2_{x_1x_2} & \sigma^2_{x_1y_2} & \sigma^2_{x_1x_3} & \sigma^2_{x_1y_3} & \sigma^2_{x_1x_4} & \sigma^2_{x_1y_4} & \sigma^2_{x_1x_5} & \sigma^2_{x_1y_5} \\ \sigma^2_{x_1y_1} & \sigma^2_{y_1y_1} & \sigma^2_{y_1x_2} & \sigma^2_{y_1y_2} & \sigma^2_{y_1x_3} & \sigma^2_{y_1y_3} & \sigma^2_{y_1x_4} & \sigma^2_{y_1y_4} & \sigma^2_{y_1x_5} & \sigma^2_{y_1y_5} \\ \sigma^2_{x_1x_2} & \sigma^2_{y_1x_2} & \sigma^2_{x_2x_2} & \sigma^2_{x_2y_2} & \sigma^2_{x_2x_3} & \sigma^2_{x_2y_3} & \sigma^2_{x_2x_4} & \sigma^2_{x_2y_4} & \sigma^2_{x_2x_5} & \sigma^2_{x_2y_5} \\ \sigma^2_{x_1y_2} & \sigma^2_{y_1y_2} & \sigma^2_{x_2y_2} & \sigma^2_{y_2y_2} & \sigma^2_{y_2x_3} & \sigma^2_{y_2y_3} & \sigma^2_{y_2x_4} & \sigma^2_{y_2y_4} & \sigma^2_{y_2x_5} & \sigma^2_{y_2y_5} \\ \sigma^2_{x_1x_3} & \sigma^2_{y_1x_3} & \sigma^2_{x_2x_3} & \sigma^2_{y_2x_3} & \sigma^2_{x_3x_3} & \sigma^2_{x_3y_3} & \sigma^2_{x_3x_4} & \sigma^2_{x_3y_4} & \sigma^2_{x_3x_5} & \sigma^2_{x_3y_5} \\ \sigma^2_{x_1y_3} & \sigma^2_{y_1y_3} & \sigma^2_{x_2y_3} & \sigma^2_{y_2y_3} & \sigma^2_{x_3y_3} & \sigma^2_{y_3y_3} & \sigma^2_{y_3x_4} & \sigma^2_{y_3y_4} & \sigma^2_{y_3x_5} & \sigma^2_{y_3y_5} \\ \sigma^2_{x_1x_4} & \sigma^2_{y_1x_4} & \sigma^2_{x_2x_4} & \sigma^2_{y_2x_4} & \sigma^2_{x_3x_4} & \sigma^2_{y_3x_4} & \sigma^2_{x_4x_4} & \sigma^2_{x_4y_4} & \sigma^2_{x_4x_5} & \sigma^2_{x_4y_5} \\ \sigma^2_{x_1y_4} & \sigma^2_{y_1y_4} & \sigma^2_{x_2y_4} & \sigma^2_{y_2y_4} & \sigma^2_{x_3y_4} & \sigma^2_{y_3y_4} & \sigma^2_{x_4y_4} & \sigma^2_{y_4y_4} & \sigma^2_{y_4x_5} & \sigma^2_{y_4y_5} \\ \sigma^2_{x_1x_5} & \sigma^2_{y_1x_5} & \sigma^2_{x_2x_5} & \sigma^2_{y_2x_5} & \sigma^2_{x_3x_5} & \sigma^2_{y_3x_5} & \sigma^2_{x_4x_5} & \sigma^2_{y_4x_5} & \sigma^2_{x_5x_5} & \sigma^2_{x_5y_5} \\ \sigma^2_{x_1y_5} & \sigma^2_{y_1y_5} & \sigma^2_{x_2y_5} & \sigma^2_{y_2y_5} & \sigma^2_{x_3y_5} & \sigma^2_{y_3y_5} & \sigma^2_{x_4y_5} & \sigma^2_{y_4y_5} & \sigma^2_{x_5y_5} & \sigma^2_{y_5y_5} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 45 & -30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -30 & 30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & -10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -10 & 70 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 80 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 40 & -13 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -13 & 60 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 30 & -25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -25 & 30 \end{bmatrix}$$

Fit a line given the observation equation:

$$mx + b = y$$

With residuals for each observation:

$$F: \qquad m(x + v_x) + b - (y + v_y) = 0$$

$$\frac{\partial F}{\partial v_x} = m$$
$$\frac{\partial F}{\partial v_y} = -1$$

Solving for Partial Derivatives:

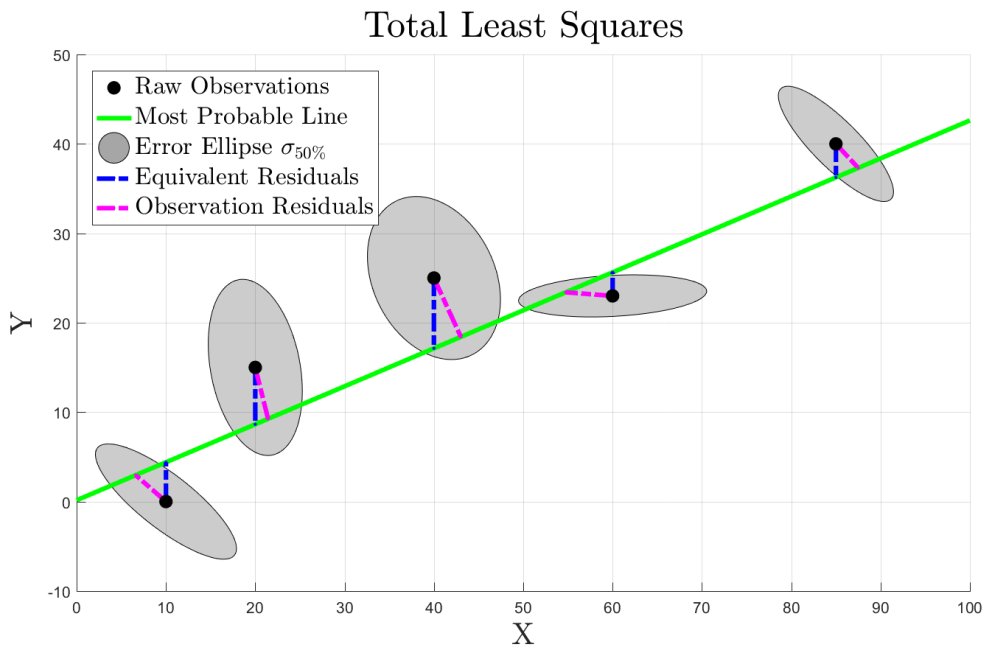
$$B = \begin{bmatrix} \frac{\partial F}{\partial v_{x_1}} & \frac{\partial F}{\partial v_{y_1}} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \frac{\partial F}{\partial v_{x_2}} & \frac{\partial F}{\partial v_{y_2}} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\partial F}{\partial v_{x_5}} & \frac{\partial F}{\partial v_{y_5}} \end{bmatrix} = \begin{bmatrix} m & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & m & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & m & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & m & -1 \end{bmatrix}$$

$$V = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \\ v_{x_3} \\ v_{y_3} \\ v_{x_4} \\ v_{y_4} \\ v_{x_5} \\ v_{y_5} \end{bmatrix} \qquad J_i = \begin{bmatrix} \frac{\partial F_1}{\partial m} & \frac{\partial F_1}{\partial b} \\ \frac{\partial F_2}{\partial m} & \frac{\partial F_2}{\partial b} \\ \frac{\partial F_3}{\partial m} & \frac{\partial F_3}{\partial b} \\ \frac{\partial F_4}{\partial m} & \frac{\partial F_4}{\partial b} \\ \frac{\partial F_5}{\partial m} & \frac{\partial F_5}{\partial b} \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \\ x_5 & 1 \end{bmatrix} \qquad \Delta X = \begin{bmatrix} \Delta m \\ \Delta b \end{bmatrix}$$

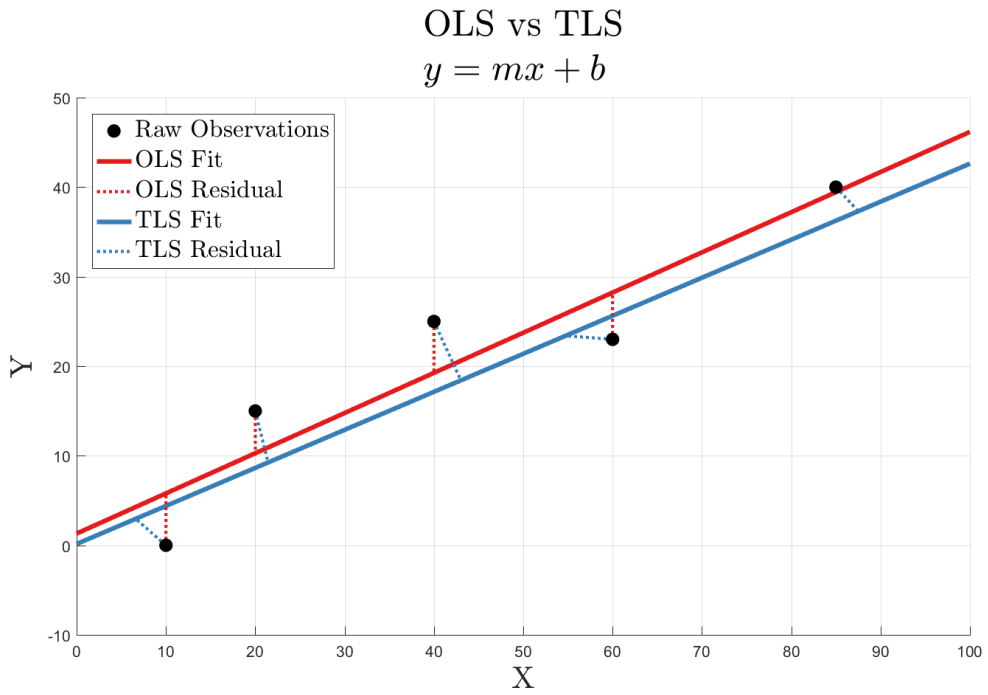
$$K_i = \begin{bmatrix} l_1 - F_1(X_i) \\ l_2 - F_2(X_i) \\ \vdots \\ l_m - F_m(X_i) \end{bmatrix} = \begin{bmatrix} 0 - (m_i x_1 + b_i - y_1) \\ 0 - (m_i x_2 + b_i - y_2) \\ 0 - (m_i x_3 + b_i - y_3) \\ 0 - (m_i x_4 + b_i - y_4) \\ 0 - (m_i x_5 + b_i - y_5) \end{bmatrix} \qquad \text{Initial Guess } X_0 = \begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

Solve Using Equations:

$n = 2$	$m = 5$	$dof = 3$
$\hat{X} = \begin{bmatrix} 0.42 \\ 0.15 \end{bmatrix}$	$V_{eq} = \begin{bmatrix} -4.39 \\ 6.36 \\ -2.63 \\ 7.86 \\ 3.75 \end{bmatrix}$	$V = \begin{bmatrix} -3.39 \\ 2.95 \\ 1.43 \\ -5.75 \\ -5.25 \\ 0.40 \\ 3.01 \\ -6.58 \\ 2.50 \\ -2.69 \end{bmatrix}$
$\Sigma_{xx} = \begin{bmatrix} 0.01 & -0.60 \\ -0.60 & 36.87 \end{bmatrix}$	$\sigma_{\hat{X}} = \begin{bmatrix} 0.11 \\ 6.07 \end{bmatrix}$	$\Sigma_{\hat{l}\hat{l}} = \begin{bmatrix} 26.05 & 21.22 & 1.92 & 11.57 & -10.14 \\ 21.22 & 17.57 & 2.94 & 10.25 & -6.20 \\ 1.92 & 2.94 & 7.01 & 4.98 & 9.56 \\ 11.57 & 10.25 & 4.98 & 7.62 & 1.68 \\ -10.14 & -6.20 & 9.56 & 1.68 & 19.41 \end{bmatrix}$
$Q_{xx} = \begin{bmatrix} 0.02 & -0.78 \\ -0.78 & 48.19 \end{bmatrix}$	$\hat{L} = \begin{bmatrix} 4.39 \\ 8.64 \\ 25.63 \\ 17.14 \\ 36.25 \end{bmatrix}$	$S_0^2 = 0.76 \qquad RMSE = 5.34$



Notice how the observation residuals are not just in the y dimension. OLS is calculated with the same data, and plotted below for comparison.



1.13.5 Example Matlab Code

Algorithm 1.9: exampleTLS.m

```
%% Example TLS Script
%% Input data
x = [10 20 60 40 85];
y = [0 15 23 25 40];
S = blkdiag([45 -30;-30 30],[20 -10;-10 70],[80 4;4 4],[40 -13;-13 60],[30 -25;-25 30]);

%% Solve TLS
Xo =[0.5; 0];
X = Xo;                      % initial unknowns guess

So2 = inf; dSo2 = 1; iter = 0;     % initialize for while loop

m = numel(x);                      % number of observations
n = numel(X);                      % number of unknowns
dof = m-n;                         % degrees of freedom
while dSo2>0 && iter<100 %loop until So2 increases or exceed 100 iteration
    B = kron(eye(numel(y)),[X(1) -1]); %B
    J = [x(:) ones(size(x(:)))];   % Jacobian
    K = -(X(1)*x(:) + X(2) - y(:));% K Matrix
    Weq = inv(B*S*B');
    dX = (J'*Weq*J)\J'*Weq*K;      % Loop Delta Estimate
    X = X + dX;                    % Loop Estimate
    Veq = K;                       % Residuals
    dSo2 = So2 - Veq'*Weq*Veq/dof; % Change in Reference Variance
    So2 = (Veq'*Weq*Veq)/dof;      % Reference Variance
    iter = iter + 1;
end

V = S * B' * Weq * Veq;           % Observation Residuals
Q = inv(J'*Weq*J);                % cofactor
Sx = So2 * Q;                     % covariance of unknowns
Sl = J * Sx * J';                 % covariance of observations
stdX = sqrt(diag(Sx));            % std of solved unknowns
Lhat = J * X;                     % predicted L values
```

Note: Matlab does not have a built in TLS function, but I wrote a function called LSRTLS.m

Algorithm 1.10: exampleTLS.m: Using lsrtls.m and anonymous function handles to perform TLS.

```

%% Example Using LSRTLS
Jfun = @(X)([x(:) ones(size(x(:)))]);
Kfun = @(X)(-(X(1)*x(:) + X(2) - y(:)));
Bfun = @(X)(kron(eye(numel(y)),[X(1) -1]));
```

1.14 Least Squares Adjustment Matlab Functions

Copying the same code over and over again, or relying on matlab toolboxes for different least squares adjustments is annoying, so here are some functions that perform the least squares calculations when provided with the input matrices and functions.

1.14.1 *lsrlin.m*: least squares regression linear

Algorithm 1.11: exampleGLS.m lsrlin.m is used to perform GLS.

```
%% Example Using LSRLIN
[X,Sx,lsainfo] = lsrlin(A,L,Sc);
```

1.14.2 *lsrnlin.m*: least squares regression nonlinear

Algorithm 1.12: exampleNonlinearLS.m lsrnlin.m and anonymous function handles are used to perform Nonlinear Least Squares.

```

%% Example Using LSRNLIN
Jfun = @(X)([sin(2*pi/2.*t + X(2)) X(1)*cos(2*pi/2.*t + X(2))]);
Kfun = @(X)(y - X(1)*sin(2*pi/2.*t + X(2)));
s = diag(stdy.^2);
```

1.14.3 *lsrtls.m*: least squares regression total least squares

Algorithm 1.13: exampleTLS.m: lsrtls.m and anonymous function handles are used to perform TLS.

```

%% Example Using LSRTLS
Jfun = @(X)([x(:) ones(size(x(:)))]);
Kfun = @(X)(-(X(1)*x(:) + X(2) - y(:)));
Bfun = @(X)(kron(eye(numel(y)),[X(1) -1]));
```

1.15 Example: TPU linear line fit

$$y = mx + b$$

TPU GLOPOV vs SLOPOV comparison of uncertainty from fit

1.16 Example: RANSAC Plane Fit

1.17 Example: 3D Conformal Transformation with TLS

Solve the 3D Conformal (7-parameter Helmert) Transformation given the following correspondences and covariance matrix.

Using the observation equations with the Tait-Bryan Rotation Matrix Convention:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos(\kappa)\cos(\phi) & \cos(\phi)\sin(\kappa) & -\sin(\phi) \\ \cos(\kappa)\sin(\omega)\sin(\phi) - \cos(\omega)\sin(\kappa) & \cos(\kappa)\cos(\omega) + \sin(\kappa)\sin(\omega)\sin(\phi) & \cos(\phi)\sin(\omega) \\ \sin(\kappa)\sin(\omega) + \cos(\kappa)\cos(\omega)\sin(\phi) & \cos(\omega)\sin(\kappa)\sin(\phi) - \cos(\kappa)\sin(\omega) & \cos(\omega)\cos(\phi) \end{bmatrix}$$

add scale to these equations

$$\begin{aligned} F: &\quad r_{11}(x+v_x) + r_{12}(y+v_y) + r_{13}(z+v_z) + T_x - (X+v_X) = 0 \\ G: &\quad r_{21}(x+v_x) + r_{22}(y+v_y) + r_{23}(z+v_z) + T_y - (Y+v_Y) = 0 \\ H: &\quad r_{31}(x+v_x) + r_{32}(y+v_y) + r_{33}(z+v_z) + T_z - (Z+v_Z) = 0 \end{aligned}$$

$$J = \begin{bmatrix} \frac{\partial F_m}{\partial S} & \frac{\partial F_m}{\partial \omega} & \frac{\partial F_m}{\partial \phi} & \frac{\partial F_m}{\partial \kappa} & \frac{\partial F_m}{\partial T_x} & \frac{\partial F_m}{\partial T_y} & \frac{\partial F_m}{\partial T_z} \\ \frac{\partial G_m}{\partial S} & \frac{\partial G_m}{\partial \omega} & \frac{\partial G_m}{\partial \phi} & \frac{\partial G_m}{\partial \kappa} & \frac{\partial G_m}{\partial T_x} & \frac{\partial G_m}{\partial T_y} & \frac{\partial G_m}{\partial T_z} \\ \frac{\partial H_m}{\partial S} & \frac{\partial H_m}{\partial \omega} & \frac{\partial H_m}{\partial \phi} & \frac{\partial H_m}{\partial \kappa} & \frac{\partial H_m}{\partial T_x} & \frac{\partial H_m}{\partial T_y} & \frac{\partial H_m}{\partial T_z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial F_m}{\partial S} & \frac{\partial F_m}{\partial \omega} & \frac{\partial F_m}{\partial \phi} & \frac{\partial F_m}{\partial \kappa} & \frac{\partial F_m}{\partial T_x} & \frac{\partial F_m}{\partial T_y} & \frac{\partial F_m}{\partial T_z} \\ \frac{\partial G_m}{\partial S} & \frac{\partial G_m}{\partial \omega} & \frac{\partial G_m}{\partial \phi} & \frac{\partial G_m}{\partial \kappa} & \frac{\partial G_m}{\partial T_x} & \frac{\partial G_m}{\partial T_y} & \frac{\partial G_m}{\partial T_z} \\ \frac{\partial H_m}{\partial S} & \frac{\partial H_m}{\partial \omega} & \frac{\partial H_m}{\partial \phi} & \frac{\partial H_m}{\partial \kappa} & \frac{\partial H_m}{\partial T_x} & \frac{\partial H_m}{\partial T_y} & \frac{\partial H_m}{\partial T_z} \end{bmatrix} \qquad \Delta X = \begin{bmatrix} S \\ \omega \\ \phi \\ \kappa \\ T_x \\ T_y \\ T_z \end{bmatrix} \qquad K_i = \begin{bmatrix} 0 - F_1(X_i) \\ 0 - G_1(X_i) \\ 0 - H_1(X_i) \\ 0 - F_2(X_i) \\ 0 - G_2(X_i) \\ 0 - H_2(X_i) \\ \vdots \\ 0 - F_m(X_i) \\ 0 - G_m(X_i) \\ 0 - H_m(X_i) \end{bmatrix}$$

$$b(r) = \begin{bmatrix} \frac{\partial F}{\partial v_{xr}} & \frac{\partial F}{\partial v_{yr}} & \frac{\partial F}{\partial v_{zr}} & \frac{\partial F}{\partial v_{Xr}} & \frac{\partial F}{\partial v_{Yr}} & \frac{\partial F}{\partial v_{Zr}} \\ \frac{\partial G}{\partial v_{xr}} & \frac{\partial G}{\partial v_{yr}} & \frac{\partial G}{\partial v_{zr}} & \frac{\partial G}{\partial v_{Xr}} & \frac{\partial G}{\partial v_{Yr}} & \frac{\partial G}{\partial v_{Zr}} \\ \frac{\partial H}{\partial v_{xr}} & \frac{\partial H}{\partial v_{yr}} & \frac{\partial H}{\partial v_{zr}} & \frac{\partial H}{\partial v_{Xr}} & \frac{\partial H}{\partial v_{Yr}} & \frac{\partial H}{\partial v_{Zr}} \end{bmatrix} \qquad B = \begin{bmatrix} b(1) & 0 & \dots & 0 \\ 0 & b(2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & b(m) \end{bmatrix} \qquad V = \begin{bmatrix} v_{x_1} \\ v_{y_1} \\ v_{z_1} \\ v_{X_1} \\ v_{Y_1} \\ v_{Z_1} \\ v_{x_2} \\ v_{y_2} \\ v_{z_2} \\ v_{X_2} \\ v_{Y_2} \\ v_{Z_2} \\ \vdots \\ v_{x_m} \\ v_{y_m} \\ v_{z_m} \\ v_{X_m} \\ v_{Y_m} \\ v_{Z_m} \end{bmatrix}$$

1.17.1 J Partial Derivatives

$$\frac{\partial F}{\partial S} = x\cos(\phi)\cos(\kappa) - z\sin(\phi) + y\cos(\phi)\sin(\kappa)$$
$$\frac{\partial F}{\partial \omega} = 0$$
$$\frac{\partial F}{\partial \phi} = -S(z\cos(\phi) + x\cos(\kappa)\sin(\phi) + y\sin(\phi)\sin(\kappa))$$
$$\frac{\partial F}{\partial \kappa} = S\cos(\phi)(y\cos(\kappa) - x\sin(\kappa))$$
$$\frac{\partial F}{\partial T_x} = 1$$
$$\frac{\partial F}{\partial T_y} = 0$$
$$\frac{\partial F}{\partial T_z} = 0$$
$$\frac{\partial G}{\partial S} = y(\cos(\omega)\cos(\kappa) + \sin(\omega)\sin(\phi)\sin(\kappa)) - x(\cos(\omega)\sin(\kappa) - \cos(\kappa)\sin(\omega)\sin(\phi)) + z\cos(\phi)\sin(\omega)$$
$$\frac{\partial G}{\partial \omega} = Sx(\sin(\omega)\sin(\kappa) + \cos(\omega)\cos(\kappa)\sin(\phi)) - Sy(\cos(\kappa)\sin(\omega) - \cos(\omega)\sin(\phi)\sin(\kappa)) + Sz\cos(\omega)\cos(\phi)$$
$$\frac{\partial G}{\partial \phi} = S\sin(\omega)(x\cos(\phi)\cos(\kappa) - z\sin(\phi) + y\cos(\phi)\sin(\kappa))$$
$$\frac{\partial G}{\partial \kappa} = -Sx(\cos(\omega)\cos(\kappa) + \sin(\omega)\sin(\phi)\sin(\kappa)) - Sy(\cos(\omega)\sin(\kappa) - \cos(\kappa)\sin(\omega)\sin(\phi))$$
$$\frac{\partial G}{\partial T_x} = 0$$
$$\frac{\partial G}{\partial T_y} = 1$$
$$\frac{\partial G}{\partial T_z} = 0$$
$$\frac{\partial H}{\partial S} = x(\sin(\omega)\sin(\kappa) + \cos(\omega)\cos(\kappa)\sin(\phi)) - y(\cos(\kappa)\sin(\omega) - \cos(\omega)\sin(\phi)\sin(\kappa)) + z\cos(\omega)\cos(\phi)$$
$$\frac{\partial H}{\partial \omega} = Sx(\cos(\omega)\sin(\kappa) - \cos(\kappa)\sin(\omega)\sin(\phi)) - Sy(\cos(\omega)\cos(\kappa) + \sin(\omega)\sin(\phi)\sin(\kappa)) - Sz\cos(\phi)\sin(\omega)$$
$$\frac{\partial H}{\partial \phi} = S\cos(\omega)(x\cos(\phi)\cos(\kappa) - z\sin(\phi) + y\cos(\phi)\sin(\kappa))$$
$$\frac{\partial H}{\partial \kappa} = Sx(\cos(\kappa)\sin(\omega) - \cos(\omega)\sin(\phi)\sin(\kappa)) + Sy(\sin(\omega)\sin(\kappa) + \cos(\omega)\cos(\kappa)\sin(\phi))$$
$$\frac{\partial H}{\partial T_x} = 0$$
$$\frac{\partial H}{\partial T_y} = 0$$
$$\frac{\partial H}{\partial T_z} = 1$$

1.17.2 B Partial Derivatives

$$\frac{\partial F}{\partial v_{xr}} = S\cos(\phi)\cos(\kappa)$$
$$\frac{\partial F}{\partial v_{yr}} = S\cos(\phi)\sin(\kappa)$$
$$\frac{\partial F}{\partial v_{zr}} = -S\sin(\phi)$$
$$\frac{\partial F}{\partial v_{Xr}} = -1$$
$$\frac{\partial F}{\partial v_{Yr}} = 0$$
$$\frac{\partial F}{\partial v_{Zr}} = 0$$
$$\frac{\partial G}{\partial v_{xr}} = S\cos(\kappa)\sin(\omega)\sin(\phi) - S\cos(\omega)\sin(\kappa)$$
$$\frac{\partial G}{\partial v_{yr}} = S(\cos(\omega)\cos(\kappa) + \sin(\omega)\sin(\phi)\sin(\kappa))$$
$$\frac{\partial G}{\partial v_{zr}} = S\cos(\phi)\sin(\omega)$$
$$\frac{\partial G}{\partial v_{Xr}} = 0$$
$$\frac{\partial G}{\partial v_{Yr}} = -1$$
$$\frac{\partial G}{\partial v_{Zr}} = 0$$
$$\frac{\partial H}{\partial v_{xr}} = S(\sin(\omega)\sin(\kappa) + \cos(\omega)\cos(\kappa)\sin(\phi))$$
$$\frac{\partial H}{\partial v_{yr}} = S\cos(\omega)\sin(\phi)\sin(\kappa) - S\cos(\kappa)\sin(\omega)$$
$$\frac{\partial H}{\partial v_{zr}} = S\cos(\omega)\cos(\phi)$$
$$\frac{\partial H}{\partial v_{Xr}} = 0$$
$$\frac{\partial H}{\partial v_{Yr}} = 0$$
$$\frac{\partial H}{\partial v_{Zr}} = -1$$

1.17.3 K Values

$$\begin{aligned}
0 - F_1(X_i) &= X_1 - tx_i + S_i z_1 \sin(\phi_i) - S_i x_1 \cos(\phi_i)\cos(\kappa_i) - S_i y_1 \cos(\phi_i)\sin(\kappa_i) \\
0 - G_1(X_i) &= Y_1 - ty_i + S_i x_1(\cos(\omega_i)\sin(\kappa_i) - \cos(\kappa_i)\sin(\omega_i)\sin(\phi_i)) \\
&\quad - S_i y_1(\cos(\omega_i)\cos(\kappa_i) + \sin(\omega_i)\sin(\phi_i)\sin(\kappa_i)) - S_i z_1 \cos(\phi_i)\sin(\omega_i) \\
0 - H_1(X_i) &= Z_1 - tz_i - S_i x_1(\sin(\omega_i)\sin(\kappa_i) + \cos(\omega_i)\cos(\kappa_i)\sin(\phi_i)) \\
&\quad + S_i y_1(\cos(\kappa_i)\sin(\omega_i) - \cos(\omega_i)\sin(\phi_i)\sin(\kappa_i)) - S_i z_1 \cos(\omega_i)\cos(\phi_i)
\end{aligned}$$

1.17.4 If Needed: Calculate Initial Estimate

Good reference: https://www.researchgate.net/publication/228608056_3-D_Coordinate_Transformations

1. Estimate scale using the standard deviation of the distance from each point to the centroid

$$Scale_0 = \frac{std(\sqrt{(x-\bar{x})^2+(y-\bar{y})^2+(z-\bar{z})^2})}{std(\sqrt{(X-\bar{X})^2+(Y-\bar{Y})^2+(Z-\bar{Z})^2})}$$

2. Pick 3 random corresponding points and estimate the rotation matrix between them

3. Apply the estimated rotation matrix to the coordinates and calculate the change in centroid

1.17.5 Matlab Example

Algorithm 1.14: example3Dconformal.m: See the full m file for J,B, and K functions

```
%example3DConformal
x = [1094.883 503.891 2349.343 1395.320];
y = [820.085 1598.698 207.658 1348.853];
z = [109.821 117.685 151.387 215.261];
X = [10037.810 10956.680 8780.080 10185.800];
Y = [5262.090 5128.170 4840.290 4700.210];
Z = [772.040 783.000 782.620 851.320];

sx = [7 11 6 5]*0.001;
sy = [8 8 5 8]*0.001;
sz = [5 9 7 9]*0.001;
sX = [5 4 2 3]*0.01;
sY = [6 6 4 5]*0.01;
sZ = [5 9 2 3]*0.01;

w = zeros(numel(x)*6,1);
w(1:6:end)=sx;
w(2:6:end)=sy;
w(3:6:end)=sz;
w(4:6:end)=sX;
w(5:6:end)=sY;
w(6:6:end)=sZ;
s = diag(w.^2); %square because std -> var for covariance matrix

Xo = calcXoEst(x,y,z,X,Y,Z); %calculate initial guess

Jfun = @(Xi)conformal3DJfun(Xi(1),Xi(2),Xi(3),Xi(4),Xi(5),Xi(6),Xi(7),...
                            x,y,z,X,Y,Z,logical([1 1 1 1 1 1 1]));

Bfun = @(Xi)conformal3DBfun(Xi(1),Xi(2),Xi(3),Xi(4),Xi(5),Xi(6),Xi(7),...
                            x,y,z,X,Y,Z);

Kfun = @(Xi)conformal3DKfun(Xi(1),Xi(2),Xi(3),Xi(4),Xi(5),Xi(6),Xi(7),...
                            x,y,z,X,Y,Z);

[X1,Sx1,lsainfo1] = lsrtls(Jfun,Kfun,Bfun,Xo,s);
```

1.18 Example: Space Resection

A space resection is used to calculate the camera pose, or exterior orientation, using correspondences between pixel coordinates and world coordinates, as well as a known camera interior orientation.

Tait-Bryan Rotation Matrix Convention (ZYX)

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos(\kappa)\cos(\phi) & \cos(\phi)\sin(\kappa) & -\sin(\phi) \\ \cos(\kappa)\sin(\omega)\sin(\phi) - \cos(\omega)\sin(\kappa) & \cos(\kappa)\cos(\omega) + \sin(\kappa)\sin(\omega)\sin(\phi) & \cos(\phi)\sin(\omega) \\ \sin(\kappa)\sin(\omega) + \cos(\kappa)\cos(\omega)\sin(\phi) & \cos(\omega)\sin(\kappa)\sin(\phi) - \cos(\kappa)\sin(\omega) & \cos(\omega)\cos(\phi) \end{bmatrix}$$

1.18.1 Computer Vision Nomenclature Equations

$s =$ Z Distance (perpendicular to camera focal plane) from Camera to XYZ world coordinate
$u =$ Camera x pixel
$v =$ Camera y pixel
$K =$ Camera Interior Orientation
$f_x =$ x focal length in pixels
$f_y =$ y focal length in pixels
$c_x =$ x principal point
$c_y =$ y principal point
$[RT] =$ Camera Exterior Orientation
$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} =$ Homogeneous World Coordinate

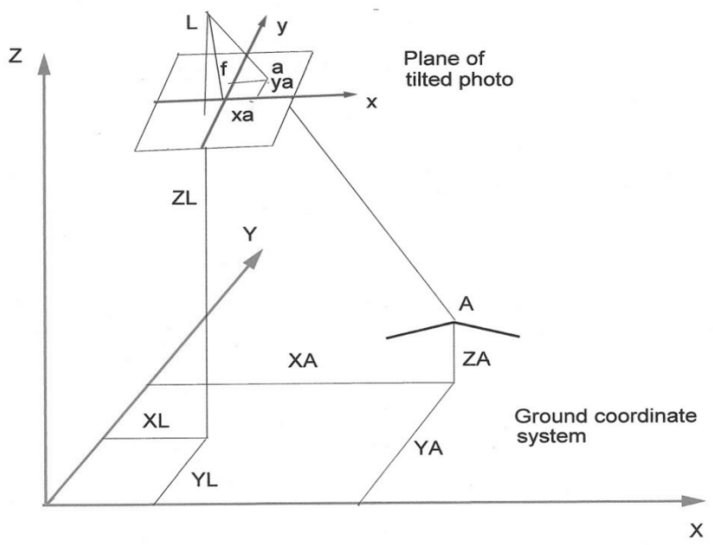
$$s \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} RT \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$s \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$(\omega, \phi, \kappa, T_x, T_y, T_z, s) =$ Space Resection Unknowns *Note that s is related to the rotation, so

1.18.2 Photogrammetry Nomenclature: "Collinearity Equations"

The photogrammetry nomenclature is slightly different, but as you can see, the equations are the same.



X_A, Y_A, Z_A = World Coordinates
X_L, Y_L, Z_L = Camera Position in World Coordinates

$$s \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_A - X_L \\ Y_A - Y_L \\ Z_A - Z_L \end{bmatrix}$$

Collinearity Equations:

above equation is wrong compared to collinearity...

$$x = x_0 - f \left[\frac{r_{11}(X_A - X_L) + r_{12}(Y_A - Y_L) + r_{13}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right]$$
$$y = y_0 - f \left[\frac{r_{21}(X_A - X_L) + r_{22}(Y_A - Y_L) + r_{23}(Z_A - Z_L)}{r_{31}(X_A - X_L) + r_{32}(Y_A - Y_L) + r_{33}(Z_A - Z_L)} \right]$$

$(\omega, \phi, \kappa, X_L, Y_L, Z_L)$ = Space Resection Unknowns

There are 6 unknowns which must be solved to determine the camera exterior orientation. For every 3D Coordinate with a corresponding pixel coordinate, there are 2 observation. Therefore, for an over-constrained solution, you need at least 4 world coordinates. *note: these points

1.19 Example: TPU Ortho From Space Resection

1.20 Example: Space Intersection

1.21 Example: SfM

1.22 References