The GoSurvey Least-Squares Adjustment Work In Progress User Guide and Mathematical Reference

Chet Jones
GoSurvey Project

October 2026

Contents

1 What the Traverse Editor adjusts **2**

2 Using the editor **3**

2.1 Getting data in . . . 3

2.2 Running an adjustment . . . 3

2.3 Project Settings . . . 3

2.4 The Output Report . . . 4

3 The command language **4**

3.1 Syntax basics . . . 4

3.2 Header directives . . . 5

3.3 Leg directives and the station chain . . . 5

3.4 Side shots and check shots . . . 6

3.5 Per-leg instrument overrides . . . 7

3.6 Network directives . . . 7

3.7 Writing, editing, and fixing the text . . . 8

3.8 A short example . . . 9

4 The observation model **9**

5 Weighting the observations **10**

5.1 Centering error 10

6 Curvature and refraction **10**

7 The functional model **11**

7.1 GPS vectors 12

8 The iterative solution **12**

9 Post-adjustment statistics **13**

9.1 Residuals and the standard error of unit weight 13

9.2 Per-measurement residual listing 13

9.3 Chi-square goodness-of-fit test 14

9.4 Error ellipses 14

10 Worked example: a square loop with an injected distance error **15**

10.1 Setup 15

10.2 Unadjusted misclosure, by hand 15

10.3 The adjustment 15

10.4 Chi-square critical values 16

11 Limitations **16**

1 What the Traverse Editor adjusts

The Traverse Editor reduces and adjusts three related kinds of jobs:

1. **A closed-loop traverse** — a chain of occupied/foresight *legs* ("courses") that starts at a known station on a known reference bearing and closes back onto that same station, possibly re-observed under a different name at the end.
2. **An open traverse tied to a known endpoint** — the same chain, but ending at a second, independently known station instead of looping back to the start. A plain open chain tied

only at the start has no redundant measurements and cannot be adjusted; one tied at both ends can.

3. **A free-form network** — any set of control points, angles, distances, azimuths, elevation differences, zenith angles, and direction sets, named directly by station rather than walked as a chain. A traverse chain and a network can appear in the same job: the chain is folded in as its own angles, distances, and starting azimuth.

Any of the three may additionally carry **GPS vectors** — baseline observations between two stations resolved from satellite positioning data, each with its own uncertainty — which are folded into the same adjustment rather than treated as fixed, unadjusted points (Section 7.1).

2 Using the editor

2.1 Getting data in

Two ways feed the same underlying job, plus a set of importers that feed one of those two:

- **Command tab** — GoSurvey's own traverse command language: a plain text format you write or edit directly. This is the primary, fully-capable editing surface, and the only one that currently represents every kind of job this guide covers (a free-form network, per-leg instrument overrides, and so on). Section 3 documents it in full.
- **Form tab** — one row per leg: horizontal angle, horizontal distance, and (optionally) a zenith angle or elevation difference. **Known limitation (work in progress):** the Form tab does not currently populate from the Command tab — neither from text you type there yourself nor from an imported file — so a job entered or edited on the Command tab, or brought in through an importer, will not show up correctly on the Form tab. The Form tab's layout is also due for a redesign. Until both are done, treat the Command tab as the one place to enter, edit, and review a job; do not rely on the Form tab to reflect what is actually in the traverse.
- **Importers** — several common field-data file formats all convert into Command-tab text, so an imported job is editable exactly like one typed by hand.

2.2 Running an adjustment

The *Closure Analysis* window's Residuals tab runs the adjustment and shows:

- the per-leg (or per-network-observation) residual table (Section 9.2);
- the GPS Vector Residuals table, when any vector was folded in;
- the chi-square goodness-of-fit result (Section 9.3);
- an **Accept Least-Squares Result** button, which writes the adjusted coordinates (and, when GPS vectors were used, the adjusted elevation) back into the legs.

If the job cannot be adjusted — too few legs, no start bearing, unresolved legs, too few redundant measurements, or a geometry the solver cannot resolve — the window reports why and shows no adjusted values; it never silently produces a meaningless result.

2.3 Project Settings

The *Project Settings...* dialog configures one project-default instrument and the solver's run options, which are saved with the job:

Field	Meaning
3D adjustment	Informational only (shown in the report); the solver still decides per-adjustment whether elevation is actually solved, from which observations are present.
Convergence limit	The solver's iterative refinement stops early once every unknown's correction in a pass is below this value (feet for a coordinate, radians for a direction-set orientation).
Max iterations	Hard cap on refinement passes regardless of convergence.
Coefficient of refraction k	Used in the curvature and refraction correction applied to a zenith observation's computed value before comparison with the field-observed zenith (Section 6).
Azimuth/bearing σ	Standard error of an azimuth or bearing observation, arc-seconds.
Zenith σ	Standard error of a zenith-angle observation, arc-seconds.
Differential level σ	Standard error of a level run's elevation difference, feet — applied as a flat value, not scaled by run length.
Centering: Instrument/Target/Vertical	A-priori centering error of the instrument over its station, the target (prism/rod) over the sighted station, and the target's vertical centering, each in feet (Section 5.1).

Changing any field recomputes the adjustment the next time results are shown.

2.4 The Output Report

The Output Report opens with a "Summary of Files Used and Option Settings" section — every file imported into this traverse, in import order; the project option settings; and the project-default instrument standard errors — ahead of the per-measurement residual listing, which itself ends with an Adjusted Coordinates section. It exports, with identical content, to:

- a plain-text file (fixed-width columns, no markup);
- a vector PDF file;
- a rich-text (RTF) file that opens natively in common word processors.

3 The command language

This section is the complete reference for the Command tab's text format — every directive, how a leg is assembled from several lines, side and check shots, per-leg instrument overrides, the network directives, and how the editor helps you write and fix the text. As noted in Section 2.1, this is currently the one place to enter and edit a job with full fidelity.

3.1 Syntax basics

The text is line-oriented and case-insensitive. Each non-blank line is one directive: a keyword, followed by its values separated by plain spaces or tabs. There is no `key=value` punctuation and no line terminator — one directive per line, values in a fixed order for that keyword.

- **Comments.** A line starting with `//` is a comment. A comment copied in from an imported file keeps its original place in the text, interleaved with the measurements around it; a blank line in that file is kept as a blank line too.
- **Field codes.** A measurement line that carried a code or description in the field ends with an apostrophe and the rest of the line, for example `HDIST 4 5 212.33 'FENCE CORNER`. The apostrophe is optional on a line you type yourself.
- **Inline options.** A line starting with `#` changes how the surrounding text is read rather than adding a measurement — for example `#LOOP ON` or `#ERRORS 5 0.02 2`. When you import a file, a marker line notes where that import begins; it does not add a measurement either.

3.2 Header directives

Written once, before any leg or network lines:

Directive	Meaning
`STATION <id> <e> <n> <z>`	The starting station's point number and coordinates.
`BEARING <deg>`	Reference orientation at the start station, decimal degrees clockwise from north. Leave this line out if there is no reference bearing yet — the job just cannot be adjusted or reduced until it is added.
`ENDPOINT <e> <n> <z>`	Known coordinates of an *open* traverse's last station. An open chain tied only at the start has no redundant measurements and cannot be adjusted; adding a known end point gives it some (Section 10 uses a closed loop instead, but the same arithmetic applies). Leave this out for a closed loop.
`LOOP ON|OFF`	Whether the traverse is a closed loop back to the start station.
`SCALE ON|OFF [COMBINED|GRID]`	Whether to reduce ground distances to grid distances using the drawing's own ground-to-grid scale factor (set elsewhere, not on this line); `COMBINED` is assumed when the kind is left off.
`ERRORS <angle-sec> <dist-const> <dist-ppm>`	The traverse-wide default a-priori standard errors for a turned angle and a distance (Section 5) — superseded, field by field, by the Project Settings dialog's own defaults and by any per-leg override (Section 3.5).

3.3 Leg directives and the station chain

A traverse leg is declared with `STN`, then described by one or more of the lines below it. Every leg names the station *occupied* for it (`AT`) and the station sighted (`TO`):

Directive	Meaning
`STN <at> <to>`	Declares an ordinary leg: station `AT` is occupied, sighting foresight `TO`.
`HANG <from> <at> <to> <angle>`	The horizontal angle turned, backsighting `FROM` through `AT` to `TO`. `FROM` is informational only — it has no effect on the computed bearing, which always comes from the start bearing plus the running sum of turned angles — but for every leg after the first it should be the station occupied for the previous leg. The first leg has no modeled backsight, so any word is accepted there (conventionally the name of the real backsight mark); it is written back out as the literal `REF`.
`HDIST <at> <to> <dist>`	Horizontal distance, entered directly.
`SDIST <at> <to> <dist>`	Slope (EDM) distance, reduced to horizontal using the leg's vertical angle.
`VANG <at> <to> <angle> [ZENITH|ELEV]`	Vertical angle; `ZENITH` (90° is level) is assumed when the kind is left off.
`DESC <at> <to> <text>`	A free-text description or field note for this leg, to the end of the line.

A leg must be declared with `STN` before any `HANG`, `HDIST`, `SDIST`, `VANG`, or `DESC` line that refers to its `TO` station, but those lines can otherwise appear in any order and on any lines below that `STN` — you do not have to group them tightly or in a fixed sequence, as long as each names the same `AT`/`TO` pair. For an ordinary leg, `AT` must continue the chain: it is the previous leg's `TO`, or the start station for the very first leg. This format represents a single sequential traverse — one instrument setup leading to the next — not an arbitrary web of setups; that is what the network directives (Section 3.6) are for.

3.4 Side shots and check shots

`STN` takes an optional third word that turns an ordinary leg into a radial (non-chain) shot:

Directive	Meaning
`STN <at> <to> SIDE`	A side shot: a detail point shot from `AT`, which must already be an occupied station (the start station or any earlier `TO`) but does not have to be the current end of the chain. `TO` is the shot's own point number. The `HANG/HDIST/SDIST/VANG/DESC` lines below it work exactly as for an ordinary leg, except `HANG`'s `FROM` is never checked against anything, since a side shot has no previous leg to continue from.
`STN <at> <to> CHECK [known-e known-n]`	A check shot: a side shot flagged as a repeat or verification observation. The optional known easting/northing are coordinates you already trust, to compare the shot's computed position against; the discrepancy is reported. Everything else is identical to `SIDE`.

Side shots and check shots describe real field observations, but they are not part of the chain that the adjustment solves, and they do not appear as rows in the residual listing (Section 9.2).

3.5 Per-leg instrument overrides

A leg can carry its own equipment figures, which replace the traverse-wide or Project Settings defaults for that one leg's observations only:

Directive	Meaning
`TSERR <at> <to> <angle-sec> <dist-const> <dist-ppm>`	This leg's own equipment accuracy, overriding the project's default angle and distance standard errors (Section 5) for just this leg's angle and distance observations.
`PRISM <at> <to> <constant> <error>`	A prism constant: a real EDM distance correction, added to this leg's slope distance before reduction. Only meaningful alongside `SDIST` — a directly-entered `HDIST` has no slope-distance step to correct. The `error` value is a-priori only; a prism's rated accuracy has no deterministic correction, only an added uncertainty (Section 5.1).
`POLE <at> <to> <error>`	The pole's centering/plumb a-priori error (its rated or estimated accuracy). Like `PRISM`'s error, this is a-priori only — there is no deterministic correction for how well a pole was plumbed over a point, only an added uncertainty.

`PRISM`'s error and `POLE`'s error combine with this leg's distance standard error (from `TSERR` if given, otherwise the project default) in quadrature, the same root-sum-square combination Section 5.1 uses for the project-wide centering errors: effective standard error $= \sqrt{\sigma_d^2 + \text{prism error}^2 + \text{pole error}^2}$. None of `TSERR`, `PRISM`, or `POLE` has any effect outside the least-squares adjustment — they do not change a reduced angle, distance, or coordinate, only how much that leg's observations are trusted when the adjustment weighs them against every other leg.

3.6 Network directives

Any set of control points, angles, distances, azimuths, elevation differences, and zenith angles, named directly by station rather than walked as a chain (Section 7 covers how each is solved). A job containing any of `ANGLE`, `DIST`, `AZIMUTH`, or `EDIF` is adjusted as a network; a traverse chain in the same text is folded in as its own angles, distances, and starting azimuth.

Directive	Meaning
`CTRL <id> <e> <n>`	Easting and northing, held fixed. No elevation.
`CTRL <id> <e> <n> <e-fix> <n-fix>`	Same, with an explicit fix on each horizontal component: `!` holds it fixed, `*` leaves it free to move, or a positive number is a standard error — the coordinate is itself a weighted measurement.
`CTRL <id> <e> <n> <z> <e-fix> <n-fix> <z-fix>`	Adds an elevation, fixed the same way.
`ANGLE <at> <from> <to> <deg> [stderr-sec]`	Clockwise angle at `AT`, from `FROM` to `TO`.
`DIST <from> <to> <dist> [stderr]`	Horizontal distance.
`AZIMUTH <from> <to> <deg> [stderr-sec]`	Azimuth, decimal degrees clockwise from north.
`EDIF <from> <to> <elev-diff> [stderr]`	Elevation difference, feet, from the first station up to the second.
`ZENITH <from> <to> <deg> [stderr-sec]`	Zenith angle. 90° is level, 0° is straight up.

Every station a network observation names must already have coordinates, either from a `CTRL` line or from the traverse chain in the same job. The optional standard error on `ANGLE`/`DIST`/`AZIMUTH`/`EDIF`/`ZENITH` overrides the project default for that one observation, the same way `TSERR` does for a chain leg.

3.7 Writing, editing, and fixing the text

The Command tab applies every keystroke live — there is no separate Apply or Revert step, so the adjustment and every other tab always reflect exactly what the text currently says. A few things make hand-editing this text practical:

- **Line numbers and highlighting.** Each line is numbered, and the text is colorized by directive so a header line, a leg line, and a comment read apart from each other at a glance.
- **Autocomplete.** Typing the start of a directive's name offers a short list of matches; Tab accepts the top one.
- **Inline errors.** If the text cannot be parsed, the offending line is highlighted directly, with a short message explaining what is wrong with it — you do not have to hunt for which line a line-number error message refers to.

Adding a leg is as simple as typing a new `STN` line (continuing the chain, or marked `SIDE/CHECK`) and the measurement lines below it; removing one is deleting those lines. Reordering, renumbering a station, or changing a single value is an ordinary text edit — the next keystroke's live-apply re-derives the whole job from the edited text, so there is nothing to keep in sync by hand.

3.8 A short example

A small closed-loop job: a start station, a reference bearing, three legs, one of them carrying a field code and a per-leg instrument override, and a free-form network fragment appended after it.

```
STATION 100 5000.00 5000.00 100.00
BEARING 0.0
LOOP    ON
ERRORS  5 0.02 2

STN   100 101
HANG  REF 100 101 90.0000
HDIST 100 101 212.33   'FENCE CORNER
VANG  100 101 90.0000 ZENITH

STN   101 102
HANG  100 101 102 88.4512
SDIST 101 102 180.70
TSERR 101 102 3 0.01 1
PRISM 101 102 0.000 0.005

CTRL   200 5500.00 5600.00 ! !
CTRL   201 5620.00 5600.00 * *
DIST   200 201 120.00 0.02
ANGLE  100 200 201 44.9000 4
```

4 The observation model

A traverse chain is a sequence of *legs* (courses). Leg k ($k = 1, \ldots, m$) runs from an *occupied* station O_k to a *foresight* station F_k, where O_k is the previous leg's foresight (O_1 is the fixed start station). Each reduced leg carries:

- a **turned (horizontal) angle** α_k — the angle at O_k from the backsight (the previous occupied station, or the start reference bearing for $k = 1$) to the foresight F_k;
- a **horizontal distance** d_k, already reduced from a slope distance and a zenith angle or vertical angle;
- optionally, an **elevation difference** Δz_k, when the leg carried a real vertical angle.

A network observation names its stations directly rather than walking a chain: a control station is held fixed, left free, or weighted (a standard error, itself an observation entering the adjustment);

an angle, distance, azimuth, elevation difference, or zenith names the stations it relates the way a chain's legs do implicitly.

Throughout, azimuths are measured clockwise from north, and every *residual* is defined as

$$v = \text{adjusted} - \text{observed}, \tag{1}$$

a single sign convention used consistently for an angle (arc-seconds), a distance (feet), an elevation difference (feet), and a GPS vector component (feet).

5 Weighting the observations

Each observation enters the adjustment with an a-priori standard error that determines its weight $w = 1/\sigma^2$. The default standard errors are:

Observation	Model	Default
Turned angle	σ_α, arc-seconds (constant)	$5.0''$
Distance	$\sigma_d = \sigma_{d,\text{const}} + \sigma_{d,\text{ppm}} \times 10^{-6} d$	0.02 ft + 2 ppm
Azimuth/bearing	σ_{az}, arc-seconds	$4.0''$
Zenith	σ_z, arc-seconds	$10.0''$
Elevation difference	$\sigma_{\Delta z} = \sigma_{v,\text{const}} + \sigma_{v,\text{ppm}} \times 10^{-6} d$	0.02 ft + 2 ppm
Differential level run	σ_{lvl}, feet (flat)	0.01 ft

A leg may carry its own equipment specification (a distance accuracy figure, or a prism/pole centering figure); when present, that per-leg figure replaces the project default for that one observation — the leg is the more specific setting.

5.1 Centering error

The project-default instrument, target, and vertical centering errors (Section 2.3) widen, rather than correct, the standard error of any sighted observation — an uncertain setup is modeled as added uncertainty, not a correctable bias. For a sight of horizontal distance D, a centering error c (feet) contributes an *angular* standard error of

$$\sigma_c = \frac{c}{D} \text{ (radians)} \longrightarrow \left(\frac{c}{D}\right) \cdot \frac{180 \cdot 3600}{\pi} \text{ (arc-seconds)}. \tag{2}$$

The instrument's and target's contributions combine with the observation's own standard error by root-sum-square:

$$\sigma_{\alpha,\text{total}} = \sqrt{\sigma_\alpha^2 + \sigma_{c,\text{inst}}^2 + \sigma_{c,\text{tgt}}^2}, \tag{3}$$

and, for a zenith observation, the vertical centering error's contribution $\sigma_{c,\text{vert}}$ (the same formula, Equation 2) is folded in the same way. For a distance observation the centering errors combine directly, already in feet:

$$\sigma_{d,\text{total}} = \sqrt{\sigma_d^2 + c_{\text{inst}}^2 + c_{\text{tgt}}^2}. \tag{4}$$

A zero sight length or a zero/unset centering error contributes nothing.

6 Curvature and refraction

A zenith angle measured over a horizontal distance D is not exactly 90° for a level sight, because the sight line bends through the atmosphere and the earth curves away beneath it. The standard elevation formula for a slope distance S and observed zenith Z is

$$\Delta\text{elev} = S\cos Z + (1-k)\frac{S^2\sin^2 Z}{2R}, \tag{5}$$

where k is the coefficient of refraction and R is the earth's radius (Ghilani and Wolf, 2015). Near a level sight ($Z \approx 90°$), $S\sin Z \approx D$, so the correction term simplifies to one that no longer depends on Z:

$$\delta Z_{\text{cr}} = \frac{(1-k)\,D}{2R} \quad \text{(radians)}, \tag{6}$$

using a mean earth radius of $R = 20{,}902{,}231$ ft. This correction is *added* to the geometric zenith computed from the adjustment's current coordinates before it is compared with the field-observed zenith, so a longer sight is modeled as reading slightly *more* than its pure-geometric zenith for $k < 1$, matching the physical effect.

7 The functional model

Let station S_i have unknown or fixed coordinates (e_i, n_i, u_i). Every observation is written as a function of the stations it names, and the solver differentiates that function with respect to whichever of those coordinates are free unknowns.

Azimuth / angle. The azimuth from O to P is $\text{az}(O,P) = \text{atan2}(e_P - e_O,\ n_P - n_O)$. Its partials with respect to the two stations' coordinates are

$$\frac{\partial\,\text{az}}{\partial e_P} = \frac{n_P - n_O}{D^2}, \quad \frac{\partial\,\text{az}}{\partial n_P} = \frac{-(e_P - e_O)}{D^2}, \quad D^2 = (e_P-e_O)^2 + (n_P-n_O)^2, \tag{7}$$

and the negatives of these with respect to (e_O, n_O). A turned angle $\alpha = \text{az}(O,F) - \text{az}(O,B)$ (foresight minus backsight, wrapped into $(-\pi, \pi]$) differentiates as the sum of the two azimuth partials with opposite signs on the backsight term; an azimuth/bearing observation is the single-term case with B replaced by the fixed start reference.

Distance. $D(O,F) = \sqrt{(e_F-e_O)^2 + (n_F-n_O)^2}$, with

$$\frac{\partial D}{\partial e_F} = \frac{e_F - e_O}{D}, \quad \frac{\partial D}{\partial n_F} = \frac{n_F - n_O}{D}, \tag{8}$$

and the negatives with respect to (e_O, n_O).

Elevation difference. $\Delta z(O,F) = u_F - u_O$, partials ± 1 on the elevation term.

Zenith. With $\text{horiz} = \sqrt{(e_F - e_O)^2 + (n_F - n_O)^2}$ and $\Delta u = u_F - u_O$, the geometric zenith is $Z = \text{atan2}(\text{horiz}, \Delta u)$ (Section 6 adds the curvature and refraction correction to this before comparison). Writing $S^2 = \text{horiz}^2 + \Delta u^2$:

$$\frac{\partial Z}{\partial u_F} = \frac{-\text{horiz}}{S^2}, \qquad \frac{\partial Z}{\partial e_F} = \frac{\Delta u}{S^2} \cdot \frac{e_F - e_O}{\text{horiz}}, \qquad \frac{\partial Z}{\partial n_F} = \frac{\Delta u}{S^2} \cdot \frac{n_F - n_O}{\text{horiz}}, \tag{9}$$

again with the opposite signs at O.

Direction sets. A direction-set observation shares one unknown *orientation* θ_{set} per set: $\text{dir} = \text{az}(O, F) - \theta_{\text{set}}$, wrapped. θ_{set}'s own initial value is resolved from the set's first direction read against the approximate coordinates, then refined as an ordinary free unknown alongside every station coordinate.

7.1 GPS vectors

A GPS vector is a baseline between two stations resolved from satellite positioning data, which may carry its own full 3×3 a-priori covariance rather than a single combined accuracy figure. When a vector carries that covariance, it is folded directly into the same adjustment as a fully correlated 3×3 observation block, rather than treated as an independent easting/northing/elevation triple:

1. The raw covariance (in an earth-centered frame) is rotated into the local east-north-up frame at the vector's origin, then converted to feet^2.
2. The rotated covariance is inverted to a full (non-diagonal) weight matrix $W = \Sigma_{\text{ENU}}^{-1}$; a degenerate (non-invertible) covariance excludes that vector from the adjustment rather than guessing at a substitute.
3. The vector contributes the correlated observation-equation block

$$\ell = \begin{pmatrix} \Delta e_{\text{obs}} - \Delta e_{\text{comp}} \\ \Delta n_{\text{obs}} - \Delta n_{\text{comp}} \\ \Delta u_{\text{obs}} - \Delta u_{\text{comp}} \end{pmatrix}, \qquad A^{\mathsf{T}} W \ell \text{ and } A^{\mathsf{T}} W A \tag{10}$$

   accumulated into the same normal-equation system as every other observation, where A is the $3 \times 2u$ (or $3 \times (u_{\text{from}} + u_{\text{to}})$) block of ± 1 design coefficients on the two stations' east/north/up terms.

Once at least one vector with a usable covariance exists, *every* free station gains elevation as a third unknown, and any leg that actually carried a vertical angle contributes an elevation-difference observation too (Section 5). With no such vector, the adjustment is a purely easting/northing solution, exactly as it is without any GPS data at all. A vector may also tie in a station that is not part of the traverse loop at all, which becomes an additional free unknown resolved jointly with the rest of the network.

8 The iterative solution

Every equation in Section 7 is nonlinear in the station coordinates, so the solver linearizes about the current approximate coordinates $x^{(0)}$ and refines iteratively:

$$\ell_i = f_i(x) - f_i(x^{(0)}) \approx \sum_j \left.\frac{\partial f_i}{\partial x_j}\right|_{x^{(0)}} \delta x_j = A_i\,\delta x, \tag{11}$$

where $\ell_i = (\text{observed} - \text{computed})_i$ is the *misclosure* of observation i at the current coordinates. Collecting every observation's row and weight into A and the diagonal (or block-diagonal, for a GPS vector) weight matrix W, each pass solves the normal equations

$$N\,\delta x = t, \qquad N = A^{\mathsf{T}} W A, \qquad t = A^{\mathsf{T}} W \ell, \tag{12}$$

applies the correction $x \leftarrow x + \delta x$, and repeats with the updated coordinates until either every unknown's correction in a pass is below the convergence limit (Section 2.3) or the maximum pass count is reached. N is symmetric positive-definite for a well-posed (non-degenerate, sufficiently redundant) geometry; a degenerate geometry (too little redundancy, or a configuration of observations that leaves a coordinate undetermined) leaves N singular, and the adjustment reports that it cannot be solved rather than return a meaningless result.

9 Post-adjustment statistics

9.1 Residuals and the standard error of unit weight

Once the refinement converges, every observation's final residual is recomputed from the converged coordinates (Equation 1), and the weighted sum of squares

$$\Omega = v^{\mathsf{T}} W v = \sum_i \left(\frac{v_i}{\sigma_i}\right)^2 \tag{13}$$

gives the *standard error of unit weight* (also called the reference standard deviation, $\hat{\sigma}_0$):

$$\hat{\sigma}_0 = \sqrt{\frac{\Omega}{r}}, \qquad r = n_{\text{obs}} - u, \tag{14}$$

the redundancy r being the number of observations minus the number of unknowns. $\hat{\sigma}_0 \approx 1$ indicates the a-priori standard errors were, on the whole, realistic; materially above 1 suggests they were optimistic (or a blunder is present); materially below 1 suggests they were pessimistic.

9.2 Per-measurement residual listing

Each course contributes one angle row and one distance row; a course with a vertical observation also contributes a vertical row; each used GPS vector contributes three rows (east, north, up). Every row carries the *standardized residual*

$$v_{\text{std}} = \frac{v}{\sigma}, \tag{15}$$

and a row is **flagged** when $|v_{\text{std}}| > 3$ — a conventional 3-sigma outlier threshold. A GPS component's standard error is the square root of that component's own variance, not a combined three-dimensional figure. Angles, distances, verticals, and GPS components are then summarized separately (only a kind with at least one row appears): count, sum of squared standardized residuals, and an *error factor*

$$\text{EF}_{\text{kind}} = \sqrt{\frac{\sum v_{\text{std}}^2}{\text{count}}}. \tag{16}$$

9.3 Chi-square goodness-of-fit test

Under the null hypothesis that the a-priori standard errors are correct and the residuals are normally distributed, Ω (Equation 13) follows a χ^2 distribution with r degrees of freedom (Ghilani, 2017). This is tested two-tailed at 5%: the adjustment *passes* when

$$\chi^2_{0.025,\,r} \;\le\; \Omega \;\le\; \chi^2_{0.975,\,r}. \tag{17}$$

The report expresses both the statistic and its bounds in the same "error-factor" units as Equation 16 — the total error factor is exactly $\hat{\sigma}_0$ — by converting each critical value:

$$\text{EF}_{\text{lower}} = \sqrt{\frac{\chi^2_{0.025,\,r}}{r}}, \qquad \text{EF}_{\text{upper}} = \sqrt{\frac{\chi^2_{0.975,\,r}}{r}}. \tag{18}$$

A failure names which bound was exceeded. The critical values themselves are computed by evaluating the regularized lower incomplete gamma function

$$P(s,x) = \frac{\gamma(s,x)}{\Gamma(s)}, \qquad s = r/2,\; x = \chi^2/2, \tag{19}$$

by the standard series/continued-fraction pair for small/large x (Press et al., 2007), against the target probability.

9.4 Error ellipses

Covariance propagates from the inverted normal-equation matrix. Writing $Q_{xx} = N^{-1}$ (the cofactor matrix) at the converged solution,

$$\Sigma_{xx} = \hat{\sigma}_0^2\, Q_{xx} \tag{20}$$

is the full covariance of every free station's coordinates. For free station i, the 2×2 easting/northing block

$$\Sigma_i = \begin{pmatrix} \sigma_{EE}^2 & \sigma_{EN} \\ \sigma_{EN} & \sigma_{NN}^2 \end{pmatrix} \tag{21}$$

has eigenvalues (the standard closed form for a symmetric 2×2)

$$\lambda_{1,2} = \frac{\sigma_{EE}^2 + \sigma_{NN}^2}{2} \pm \sqrt{\left(\frac{\sigma_{EE}^2 + \sigma_{NN}^2}{2}\right)^2 - (\sigma_{EE}^2\sigma_{NN}^2 - \sigma_{EN}^2)}, \tag{22}$$

giving the 1-sigma semi-major and semi-minor axes $a = \sqrt{\lambda_1}$, $b = \sqrt{\lambda_2}$, and the major axis's azimuth from the eigenvector angle $\theta = \frac{1}{2}\,\mathrm{atan2}(2\sigma_{EN},\, \sigma_{EE}^2 - \sigma_{NN}^2)$, converted from a math angle off the easting axis to an azimuth clockwise from north. The ellipse is empty (all zero) exactly when the geometry could not support an inverse — the same condition that would make the adjustment itself unsolvable.

10 Worked example: a square loop with an injected distance error

10.1 Setup

Start station 0 at $(E, N) = (0, 0)$, reference bearing 0° (north), four legs each turning 90° to the right, nominal length 100 ft, except leg 1 (station $0 \to 1$) measured 0.10 ft long:

Leg	Occupied → Foresight	Turned angle	Horizontal distance (ft)
1	$0 \to 1$	90°	100.10
2	$1 \to 2$	90°	100.00
3	$2 \to 3$	90°	100.00
4	$3 \to 0$	90°	100.00

10.2 Unadjusted misclosure, by hand

Walking the chain with the observed angles/distances (azimuth of leg $k{+}1$ = azimuth of leg k's backsight $+\,90°$, the convention underlying Equation 7): leg 1 azimuth = $0° + 90° = 90°$ (east) ⇒ station 1 = (100.10, 0). Leg 2's backsight azimuth (1→0) is 270°, so leg 2's azimuth is $270° + 90° = 0°$ (north) ⇒ station 2 = (100.10, 100). Leg 3's backsight azimuth (2→1) is 180° (same easting), so leg 3's azimuth is $180° + 90° = 270°$ (west) ⇒ station 3 = (0.10, 100). Leg 4's backsight azimuth (3→2) is 90°, so leg 4's azimuth is $90° + 90° = 180°$ (south) ⇒ closing point = (0.10, 0).

The closing point misses the true start $(0, 0)$ by exactly

$$(\Delta E, \Delta N) = (0.10,\ 0.00)\ \text{ft}, \qquad \text{linear misclosure} = 0.10\ \text{ft}, \tag{23}$$

which is entirely attributable to leg 1's extra 0.10 ft, since every turned angle was exact.

10.3 The adjustment

With the default weights (Section 5; nearly equal distance weight on all four ∼100 ft legs since the ppm term is tiny, and equal angle weight on all four turned angles), the adjustment counts 8 observations (4 angles + 4 distances) against 6 unknowns (stations 1, 2, 3 free in two dimensions), for a redundancy of 2. Because the only error is a distance, not an angle, the adjustment drives the correction overwhelmingly into leg 1's own distance (its adjusted length comes out shorter than the 100.10 ft observed) while leaving the angles close to their observed values; the closure

recomputed from the adjusted observations comes out below 0.001 ft, three orders of magnitude tighter than the unadjusted 0.10 ft misclosure, with a finite, positive standard error of unit weight.

10.4 Chi-square critical values

A separate, exactly hand-checkable part of the goodness-of-fit test is the critical-value computation itself, which can be checked directly against published χ^2 table values (Ghilani, 2017):

Degrees of freedom r	$\chi^2_{0.025,\,r}$ (published)	$\chi^2_{0.975,\,r}$ (published)
1	0.000982	5.023886
2	0.050636	7.377759
10	3.246973	20.483177

The in-application computation of these critical values (Section 9.3) reproduces every one of these to within 0.2% (within 5% for the smallest, $r = 1$, 2.5% value, where the critical value itself is tiny and relative error is a poor metric).

11 Limitations

A few simplifications are deliberate, not oversights:

- A differential-level run's standard error (Section 5) is applied as a flat value per run, not scaled by run length, since the import formats in use here do not always carry a run length to scale by.
- There is no separately field-proven vertical (elevation-difference) error model; it reuses the same constant-plus-parts-per-million form as a horizontal distance (Section 5).
- Only one project-wide default instrument exists; it can be overridden per leg with a distance accuracy figure or a prism/pole centering figure, but not swapped wholesale per imported file.
- Side-shot observations and some imported direction-set data are not yet supported on import, independent of the adjustment's own direction-set support for hand-entered network text.

References

Ghilani, C. D. (2017). *Adjustment Computations: Spatial Data Analysis*, 6th ed. Hoboken, NJ: John Wiley & Sons.

Ghilani, C. D., and Wolf, P. R. (2015). *Elementary Surveying: An Introduction to Geomatics*, 14th ed. Boston: Pearson.

Leick, A., Rapoport, L., and Tatarnikov, D. (2015). *GPS Satellite Surveying*, 4th ed. Hoboken, NJ: John Wiley & Sons.

Mikhail, E. M., and Ackermann, F. (1976). *Observations and Least Squares*. New York: IEP–A Dun-Donnelley Publisher.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. Cambridge: Cambridge University Press.