LineTerra

User Manual

Road, dam, drainage, pad and slope-stability earthworks design for Wolfram Mathematica

Covers LineTerra v1.1, including DXF export of sections and plan

Contents

1 Overview

LineTerra is a single-file Wolfram Language package (`LineTerra.wl`) for designing and quantifying earthworks. Its core method is to thread a cross-section template along an alignment and calculate its intersection with the existing ground; the template can be as complicated as wished, and any number of templates can be assigned along the route. Output includes plans, longitudinal and cross sections, quantities and charts – including a mass-haul diagram – and 3D views. The program is highly graphical and interactive: horizontal alignment (intersection points with circular or Bezier curves) and vertical profile (grade points with parabolic K-value curves) moves are reflected immediately in the charts, so a design can be fine-tuned quickly to minimise earthworks and cost.

Although most of this manual illustrates LineTerra by designing roads, the same procedure applies to a much wider range of earthworks tasks: obtaining a suite of cross-sections for slope-stability design, designing dams and calculating the volume of water they impound, laying out drainage networks, and pad design for a single building platform. Its worked examples range from a small pad for a single house to the preliminary design of a 160 km motorway.

In the early stages of a project, high-quality survey data is often not yet available, so, to stay useful throughout the design process, LineTerra accepts terrain from many sources: GeoTIFF, compressed LAZ or uncompressed LAS LIDAR point clouds, delimited XYZ survey files, `GeoElevationData` from Wolfram's own database, or a synthetic corridor strip – with Wolfram's Streetmap and satellite photo data available as plan backdrops. Work can be carried out in latitude/longitude or in national grid coordinates, and the grids of most countries are included (see §15). LineTerra also supports optional geological strata shown on each cross-section, multiple roads (or other alignments) sharing one piece of ground, and "baking" a finished design into the terrain so the next one can be built on top of it.

`LineTerra.wl` is distributed as an installable `.paclet` archive – see §2.

1.1 Requirements

- Wolfram Mathematica with a working Front End. The interface is built from Dynamic content, dialogs, palettes and `CreateDocument` windows, so LineTerra will not run under a bare kernel or `wolframscript` – it needs the notebook front end.
- Internet access is only needed for three optional features: fetching elevation via `GeoElevationData` (the "Lat/Lon" terrain source), the GeoMap/Aerial plan backgrounds, and downloading LIDAR tiles from a supplier. GeoTIFF, XYZ, LAZ/LAS and Corridor terrain, and Relief/Contour/Hybrid plan modes, all work offline.
- LAZ/LAS point-cloud terrain needs an external decoder – see the dedicated note in §6.2.

2 Loading LineTerra

`LineTerra-1.1.0.paclet` is a standard installable Wolfram paclet. Install it once, following the instructions in the Paclet repository, where you obtained the paclet, then load it like any other package in any future session:

```
PacletInstall["/path/to/MalcolmWoodruff/LineTerra-1.1.0.paclet"] or similar
```

Then, once per session, if you need LineTerra.

```
Needs["LineTerra`"] or Needs["MalcolmWoodruff`LineTerra`"]
```

Then for each road or time you need LineTerra

```
LineTerra["id"]
```

Where LineTerra["id"] opens a toolbar cell for a road named "id". Evaluating `LineTerra` with the same name again reopens that road's toolbar; evaluating it with a different name opens another road (see §13).

3 Using LineTerra and this manual

If this is the first time you have used LineTerra, I suggest that you run through the 6 minute road example. This shows the complete workflow and only takes 6 minutes! After which, you should have a good understanding of the program. You may then skip section 4 and look at sections 5-16, which describe each section/window of the program in detail. Then there are a number of examples using different terrains or where the program is being used for different purposes. I recommend that you study the notes in section 18 on timing, accuracy and quantity calculation so you have an understanding of the programs limitations. Which are mostly a reflection of the accuracy of the terrain presented to it. Finally, there is a quick reference guide and some ideas for using LineTerra for more than roads in section 21.

4 Package Structure and the Public API

`LineTerra.wl` is wrapped in its own context:

```
BeginPackage["LineTerra`"]
  (* ~87 public symbol declarations, one per line *)
Begin["`Private`"]
  (* the entire implementation *)
End[]
EndPackage[]
```

This keeps LineTerra's symbols out of `Global``, so they can't collide with anything already defined in your own notebook. Two consistent naming conventions tell you, at a glance, whether a symbol is meant to be called directly:

- **Public API** – Capitalised, LT-prefixed names (`LTChainages`, `LTBakeRoad`, `LTExportSections`, . . .), the two entry points `LineTerra` / `LineTerraNewProject`, the LAZ helpers `ImportLAS` / `ImportLAZ` / `LAZtoLAS`, and `$LTEarthRadius` / `$LTState`. These live in `LineTerra`` and are exactly what you'd call from your own code.
- **Private implementation** – lowercase lt-prefixed names (`ltDrawXS`, `ltComputeXS`, `ltSaveDlg`, . . .) and `$lt`-prefixed globals (`$ltXSTemplates`, `$ltTerrain`, . . .). These live in `LineTerra`Private``, are not exported, and are implementation detail that can change between versions without notice.

This split was verified by loading the wrapped file in a live kernel: a clean `Get` with no shadow or syntax warnings, exactly 87 names on `LineTerra`*`, several thousand on `LineTerra`Private`*`, and confirmed working core functions (`LTChainages`, `LineTerra`, `$LTEarthRadius`, `$LTState`) called through the public context.

5 The Toolbar

Every LineTerra window is opened from the toolbar cell that `LineTerra["name"]` prints. Buttons open a panel or window; most panels can be left open and will update live as you keep working on the plan. On a small screen the Plan can be resized and the windows stacked for convenience. A Close Views button is available to remove all subsidary windows at once, if desired.

Button	**What it does**
Close Views	Closes every LineTerra window belonging to this session (plan, sections, 3D, etc.).
+ New Project	Clears the loaded terrain and alignment and starts a genuinely empty project.
+ Add Road	Opens a second (third, ...) road on the same terrain; the existing road is kept and shown as a grey overlay on the new one.
Terrain	Terrain source panel: GeoTIFF, Lat/Lon, XYZ, LAZ/LAS, Corridor, photo overlay, strata.
Plan/Profile	The main design window: plan, vertical profile and earthwork, combined.
Intersect	Finds every crossing between this road and other stored roads and inserts a matching IP into both alignments.
Templates	Cross-section template editor and template-library manager.
Sections	Interactive cross-section viewer, stepped by chainage along the alignment.
Earthwork	Computes cut/fill quantities and overlays them on the plan.
Export	CSV / PDF / PNG / notebook export panel for every data type below.
3D View	Rotatable 3D terrain, road and cut/fill view.
Bake Road	Embeds the finished road's formation into the terrain's elevation matrix.
Unbake Road / Make Permanent	Appear once a bake exists; revert to the pre-bake terrain, or discard the undo snapshot.
Coords	Standalone lat/lon ↔ grid coordinate converter, independent of the loaded project.
Help	In-app quick-reference panel (reproduced in §20 of this manual).

6 Terrain

Every road needs a terrain before the plan, profile, templates or earthwork can do anything. Open it from the Terrain button.

6.1 Terrain sources

- **GeoTIFF** – load a `.tif` elevation raster together with its matching `.tfw` world file, as supplied by a survey or national mapping agency. If georeferencing tags are embedded in the TIFF itself, `ltReadGeoTIFFTags` reads `ModelPixelScaleTag`/`ModelTiepointTag` directly, so a `.tif` with embedded georeferencing can be read without a separate `.tfw` sidecar at all. Some GeoTIFFs – commonly ones with resolution overviews/pyramids built in by GDAL – store several pages of different sizes: the full-resolution image plus smaller preview levels for fast zoomed-out rendering. `Import` then returns a list of pages rather than a single flat 2D array, since the pages can't be stacked; pick out the full-resolution page (the largest one) in that case.
- **Lat/Lon** – fetches elevation directly from Wolfram's `GeoElevationData` for a location or bounding box. No local files needed, but requires internet.
- **XYZ file** – imports a tab/space/comma-delimited survey point file (easting, northing, elevation).
- **LAZ/LAS point cloud** – imports a compressed (`.laz`) or uncompressed (`.las`) LIDAR survey and grids the ground-classified points into terrain. See the LAZ note below – this source needs an external decoder. Once that is installed, importing LAZ files is as straightforward as importing GeoTIFF files.
- **Corridor** – builds a narrow synthetic terrain strip that follows a set of waypoints. Useful for early-stage design before full survey data exists – opening Plan/Profile on a fresh corridor terrain seeds the alignment IPs from the corridor waypoints automatically. See §17 for a worked example of a 160 km corridor.

6.2 LAZ/LAS terrain requires an external decoder

Wolfram Language has no native LAZ importer. A `.laz` file uses an adaptive arithmetic coder (LASzip); LineTerra decompresses it by delegating to an external tool, then reads the resulting `.las` file natively in WL – straight from `LineTerra.wl`'s own comments:

```
Wolfram Language has no native LAZ/LAS importer. LAZ uses an
adaptive arithmetic coder (LASzip) -- decompression is delegated
to an external decoder (PDAL or the laszip CLI); the resulting
LAS file is then read natively in WL below, no further external
dependency. Requires PDAL or LASzip on the system path, e.g.:
  brew install pdal (recommended, also handles many
                            other geospatial formats)
  brew install laszip
  conda install -c conda-forge pdal (if you use Anaconda)
If installed via conda, set $ltLAZDecoderPath to its full path
(find it with "which pdal" in an activated terminal) -- a
GUI-launched Mathematica does not see an activated conda env's
PATH, e.g.:
  $ltLAZDecoderPath =
    "/Users/malcolmwoodruff/anaconda3/envs/pdal-env/bin/pdal";
```

In practice this means: **LAZ terrain loading is not pure Wolfram Language – it relies on an external Python-distributed tool (PDAL) to decompress the LAZ file to LAS.** This is the only external dependency LineTerra has outside Mathematica/Wolfram. The two most common routes to get PDAL are Homebrew (`brew install pdal`, macOS/Linux) or a Python/conda environment (`conda install -c conda-forge pdal`). The worked LAZ example in §17 states the same requirement directly: *"you must have PDAL or LASzip on the system path, e.g.: brew install pdal."* If you install it via conda, remember that a GUI-launched Mathematica does not inherit an activated shell's PATH, so `ltFindLAZDecoder[]` will not find it automatically – run `which pdal` inside the activated conda environment in a terminal, then set:

```
$ltLAZDecoderPath = "/full/path/to/envs/pdal-env/bin/pdal";
```

before loading LAZ terrain. Uncompressed `.las` files need no external tool at all – `ImportLAS` reads the LAS binary format natively (point data formats 0–10, including LAS 1.4 extended formats).

6.3 Aerial / OS photo overlay

A georeferenced photo (aerial or OS raster) can be draped under the plan's contour lines. In most cases `GeoGraphicsOverlay` is used instead, showing a Streetmap or satellite view as the background. Use a photo overlay when you specifically want your own aerial photo or map extract – select "Photo+Contours" when previewing the terrain and you'll be asked to pick a photo file. Make sure the photo covers the same boundaries as the terrain.

6.4 Geological strata - In preparation

Named strata (soil/rock layers) can be attached to a loaded terrain with `LTAddStratum`. Each stratum is shown as a coloured band on cross-sections, and its visibility can be toggled independently from a checklist below the plan. This feature is in preparation and will be in a future release.

6.5 Plan render modes

The plan itself can be switched between Relief, Contour, Hybrid, GeoMap and Aerial rendering; GeoMap and Aerial fetch online map tiles and need internet, the other three work entirely offline from the loaded terrain data.

7 Plan & Horizontal Alignment

The Plan/Profile window is the main design surface. The Plan/Profile window will take some time to render initially, as it has to get the information that makes up the terrain. This delay only occurrs once. Its plan pane sets out the horizontal alignment:

- Click anywhere on the plan to add a new Intersection Point (IP).
- Drag an existing IP to move it – the profile, templates, sections and earthwork all recompute automatically.
- Each IP carries a radius and curve type, so the alignment rounds smoothly through it with a circular arc or Bezier curve instead of kinking.
- Delete mode toggles a click-to-remove mode for IPs.
- A zoom window can be drawn by dragging a box on the plan, opening a magnified detail palette.
- The Plan Size slider resizes the plan (the profile view below it follows).
- Contour interval can be set to None to hide contour lines on the terrain background.

Because chainage, profile, templates and earthwork are all derived from the live alignment, moving an IP after the rest of the design exists is safe – everything downstream updates rather than going stale.

7.1 Polygon / pad tool

Alongside the alignment-and-template machinery used for roads, the Plan/Profile window has a separate polygon tool for footprint-based earthworks – a building pad, a lagoon, a laydown area, or any other area best described by its boundary rather than a centreline:

- **Add Polygon** – click to start; click each boundary vertex on the plan in turn; **Stop Polygon** closes it (needs at least 3 vertices).
- **Formation RL** – the platform level, entered directly or set with ← **Mean ground**, which reads the average natural ground level inside the polygon (on the same refined grid the quantities themselves use, so the two stay consistent).
- **Vertical walls** – ticked: a flat platform with vertical faces at the boundary, quantities reported for the footprint alone. Unticked: one-sided batters run outward from the boundary like a road, quantities reported for the interior plus the batter ring.
- **Floor grade from VIPs** – unticked, the base is flat at the Formation RL. Ticked, the base instead follows a triangulated surface through Design RLs set as VIPs on the polygon's own control line (compute Earthwork rows first).
- **Refine** – resamples the terrain to a finer grid (Off / 5 m / 2 m / 1 m) before baking; see §18 for why this matters for pads specifically.
- **Bake Polygon** – embeds the pad into the terrain and reports its cut/fill quantities, the polygon equivalent of Bake Road (§12).

8 Vertical Profile

The profile pane, shown beneath the plan, carries the road's vertical design as a set of VIPs (vertical intersection points): a chainage and a design RL (level).

- Each VIP carries a K-value, which produces a parabolic vertical curve through it rather than a kinked grade line.
- Because chainage comes from the horizontal alignment, the ground profile shown beneath the design line updates live as IPs are moved in the plan above it.
- Design RL and depth of cut/fill are read directly off this profile by the cross-section and earthwork calculations.

9 Cross-Section Templates

A template describes the road's shape moving outward from the centreline, on the Left and Right sides independently (or mirrored), as a chain of links.

9.1 Links

Each link has a Width (W), Slope % (S, rise per unit width × 100 – positive is up, negative is down), Depth (D) and Colour, and a Cut flag:

- **Cut = False** (a "fixed" link) is drawn at its literal width regardless of the terrain – used for the carriageway, a kerb, a berm, a vertical wall, and similar fixed elements.
- **Cut = True** (a "batter" link) instead searches outward until it crosses the ground, up to a maximum search width (MaxW) set on the template – used for cut and fill slopes that must meet natural ground wherever that happens to be.

9.2 Presets

Preset	Typical use
Link	Plain carriageway strip – flat, fixed width.
CutBat	2:1 cut batter, searches out to meet rising ground.
FillBat	3:1 fill batter, searches out to meet falling ground.
Berm	Flat shelf, fixed width.
Wall	Near-vertical retaining element.
DitchIn / DitchOut	Inward- or outward-sloping ditch.
Split	A branch point: automatically follows the CutBat sub-links where the ground is above formation, or the FillBat sub-links where it is below – one link definition handles both cases along a corridor that alternates between cut and fill.

9.3 Template-wide settings

- **Mirror** – when on, the Left side reuses the Right-hand link list; when off, Left and Right are defined independently.
- **MaxW** – the furthest a batter link is allowed to search for a terrain crossing before giving up (a "Limit" batter).
- **TM** (Template Margin) – how far past each side's batter point the ground profile is drawn and trimmed, both for the section view and for the earthwork search width.

9.4 Structures: Bridge and Tunnel

A template can be flagged with a Structure type:

- **None** (default) – ordinary cut/fill batter behaviour, as above.
- **Bridge** – no cut/fill quantities are reported at all, since the deck spans over the ground; a deck thickness (BridgeThk) is recorded separately for deck-quantity purposes.
- **Tunnel** – the excavated area is computed from the formation width and a clearance height (TunnelClear), and reported entirely as Cut.

9.5 Assigning templates along the alignment

Templates are assigned to chainage ranges in the Sections Tab. Where the alignment moves from one assignment into a differently-shaped template, LineTerra linearly interpolates link widths and slopes across the transition rather than jumping abruptly from one shape to the other. Assignments do not need to be contiguous: leaving a gap between two templates with the same number of links lets LineTerra interpolate across it, which is how a road can be widened for bus stops or slow lanes without a separate template for every intermediate chainage.

9.6 Template libraries

A full set of templates can be saved to, loaded from, or merged from a `.ltl` file (Mathematica's native MX format) using the buttons in the Templates panel, so a library built for one project can be reused on the next.

10 Cross-Sections: Viewer & Export

The Sections window steps through the alignment chainage by chainage, drawing the ground profile, the applied template, and cut (red, terrain above formation) / fill (green, terrain below formation) shading, with batter type and offsets labelled.

10.1 Exporting a range of sections

From the Export panel's Sections row, choose a From/To chainage range, an interval, and a vertical exaggeration (VE), then use either export button:

- **PDF (to scale)** – every section in the range is exported at an exact 1:N drawing scale, in both the horizontal and vertical directions – vertical exaggeration only stretches the height, it never distorts the true horizontal scale. Set the scale denominator in the "PDF 1:" field (e.g. 500 for 1:500). Each page is labelled with its chainage and design RL, frame ticks and axis labels included.
- **Notebook** – saves a single evaluatable cell defining `crossSectionPlots`, an Association of the form "Ch = N m" → graphic. Open the file and evaluate that cell (Shift+Enter) to bind the name in your own kernel; each graphic is a genuine vector `Graphics` object (not a rasterised bitmap), sized to the "NB px" width with an automatic aspect ratio, so it can be pulled out by name, resized by dragging its corner handles, or laid out however you like, for example:

```
LayoutGrid[Partition[Values[crossSectionPlots], 3], Frame -> All]
```

- **DXF** – exports every section in the range into a single CAD-ready `.dxf` file, real-world metre coordinates throughout (no drawing-scale conversion is needed – the geometry is already true to scale). See §16.1 for the full details of what's in the file and why it's built the way it is.

11 Earthwork

The Earthwork panel computes cut and fill cross-sectional area at each chainage from the applied template against the ground, then integrates between sections to give cut and fill volume. Results are shown as a table and as a coloured overlay on the plan.

11.1 CSV export columns

Column	Meaning
Ch	Chainage (m).
CutArea / FillArea	Cross-sectional cut / fill area at this chainage (m^2).
LeftToe / RightToe	Offset from centreline where each batter meets the ground (m).
CutVol / FillVol	Cut / fill volume between this section and the previous one (m^3).
CumCut / CumFill	Running total cut / fill volume along the alignment so far (m^3).

12 Baking a Road into the Terrain

Bake Road writes the finished road's formation and batters into the terrain's own elevation matrix, so the ground looks genuinely built rather than just having a road design overlaid on the original survey. Earthwork must have been computed at least once before baking is available.

This exists for staged, multi-road design: bake Road A once it's finalised, then design Road B on the resulting terrain so its earthwork and cross-sections correctly account for Road A already being there.

- **Unbake Road** – reverts to the terrain as it stood immediately before the most recent bake. The pre-bake snapshot is kept until you bake again or click Make Permanent.
- **Make Permanent** – discards the undo snapshot; the bake can no longer be reversed, and the baked terrain becomes the new starting point.

13 Multiple Roads on One Terrain

`LineTerra["Name"]` opens a road with that name; if a terrain is already loaded, the new road inherits it, so several roads can be designed on one piece of ground without reloading terrain each time.

- **+ Add Road** opens a second (third, . . .) toolbar directly from an existing one, inheriting the same terrain.
- Every other stored road is drawn as a grey overlay on the active road's plan, so you can see how they relate spatially while designing.
- **Intersect** scans every other stored road for crossings with the active one and inserts a zero-radius IP into both alignments at each crossing point (an existing IP within 1 m of the crossing is reused rather than duplicated).
- **+ New Project** clears the terrain entirely and starts fresh; this is different from opening another road name, which keeps the terrain.

In short: design the first road as normal, then click + Add Road (or Plan/Profile on a second road name) to open a new Plan/Profile window sharing the existing terrain, with the first road shown in grey. Design the new road as usual; if the two cross, Intersect inserts IPs at every crossing.

```
LineTerra["Road A"] (* open or switch to Road A’s toolbar *)
LineTerra["Road B"] (* open Road B; Road A’s state is saved automatically *)
```

14 3D View

Opens an interactive, rotatable 3D view: the terrain surface (subsampled for performance), the road centreline draped over it in 3D, and cut sections shaded red / fill sections shaded green along the corridor.

15 Coordinate Export & Conversion

If the source terrain is already in National Grid coordinates, the plan, sections and every other output stay in those same coordinates throughout. If Lat/Lon was used instead, the plan and sections work in a local grid, but the alignment can still be exported in both Lat/Lon and a national grid – see below.

15.1 Coordinate export

From the Export panel’s Coordinate Export section, choose a coordinate reference system and a spacing, then Compute and Save CSV:

CRS	Notes
Local	The terrain’s own plan grid (metres). No datum conversion, so no terrain origin is needed – always available.
WGS84	Decimal-degree latitude/longitude. Works for any terrain type.
OSGB36	British National Grid. Requires a lat/lon terrain origin (originLL).
EPSG:...	Any UTM zone or other supported projection – enter the EPSG code directly, e.g. EPSG:25831 for Spain UTM 31N.

Spacing of 0 exports IPs only; a spacing greater than zero exports setting-out points every N metres along the curve-resolved centreline (with the end point always included), which is what a curved road actually looks like on the ground rather than just its IPs. Design RL and ground RL columns are appended automatically wherever that data is available.

15.2 Standalone converter

The Coords button on the toolbar opens a lat/lon ↔ grid converter, for any supported national grid, that works independently of any loaded project – useful for a quick one-off coordinate lookup. **This should be checked before relying on it.** For the British National Grid there is an adjustment applied to bring the grid conversion in line with Ordnance Survey’s own published values; it has been included here and should be accurate to about 80mm, but may not carry over to other countries’ grids in the same way, so check any national grid conversion against known control values before using it for real setting-out.

16 Export Reference

Every export in LineTerra is reached from the Export panel (toolbar → Export).

Data	Format	Contents
Alignment	CSV	IP, Easting, Northing, Radius, CurveType
Profile	CSV	VIP, Chainage, RL (m ODN), K-value
Earthwork	CSV	Ch, CutArea, FillArea, LeftToe, RightToe, CutVol, FillVol, CumCut, CumFill
Plan	PDF / PNG	Current plan view, at a chosen pixel size
Plan	DXF	Alignment centreline and terrain contours, plus chainage ticks and a scale bar – see §16.1
Sections	PDF	Every section in the chosen chainage range, at an exact 1:N drawing scale
Sections	Notebook (.nb)	`crossSectionPlots` – a named Association of vector section graphics
Sections	DXF	Every section in the chosen chainage range, true-scale, with a ruler layer – see §16.1
Coordinates	CSV	Setting-out coordinates in the chosen CRS, at the chosen spacing

16.1 DXF export

Both the Sections and the Plan rows of the Export panel offer a DXF button, for bringing LineTerra's geometry into AutoCAD, Civil 3D, Vectorworks or any other DXF-reading CAD package. DXF export has a few properties that are worth understanding before you rely on it.

16.1.1 Coordinates are always true scale

Unlike the PDF section export, which lets you choose a drawing scale (§16), DXF coordinates are written directly in real-world metres. There is nothing to set: 1 drawing unit is 1 metre, so measuring in your CAD package gives the true dimension directly.

16.1.2 No text labels – by design

DXF files can carry a `TEXT` entity type for labels, but LineTerra deliberately never writes one. Mathematica's own `Import[..., "DXF"]` fails the *entire file* with `$Failed` the moment it encounters a bare `TEXT` entity, even one sitting alongside otherwise perfectly valid geometry – confirmed by isolating a single `TEXT` entity in an empty file and reproducing the failure on its own. Since re-opening an exported DXF file with `Import` in Mathematica is a normal, expected workflow here (not just opening it in third-party CAD software), LineTerra avoids `TEXT` entities entirely, in both the section and plan exporters.

Two things stand in for text labels instead:

- **Layer names.** Every piece of geometry is identifiable purely by which layer it's on – `GROUND_ChN` / `DESIGN_ChN` / `SCALE_ChN` per section (`N` = the rounded chainage), and `ALIGNMENT` / `CONTOURS` / `CHAINAGE_TICKS` / `SCALE_BAR` on the plan. Any CAD layer manager lets you filter, colour-code or toggle by these names.

- **Geometric dimension marks.** A pure-geometry ruler is drawn alongside the design geometry – described below – so a chainage, offset or scale reference doesn't require a number to be printed on the drawing at all.

16.1.3 Cross-section DXF: the SCALE_ChN ruler

Each exported section carries a ruler on its own `SCALE_ChN` layer, drawn just below the lowest point of that section:

- A baseline spans the full exported width of the section.
- Minor ticks are drawn at a round interval (automatically chosen from 0.2/0.5/1/2/5/10/20/50/100 m etc. so that roughly 8–20 ticks appear across the section).
- Every 5th tick is drawn taller (a "major" tick), the same convention as a paper scale bar.
- A taller tick marks the centreline (offset 0).
- The tallest ticks mark the section's *true* left and right toe – the actual point where each batter meets the ground, before the export margin is added – so the width that actually controls the design is identifiable by geometry alone, without needing a printed number.

The exported ground and design (batter) lines are clipped to this same true-toe-plus-margin width, so a DXF section always matches what the PDF export and the on-screen Sections viewer show for the same chainage – not the batter's raw, uncorrected maximum search width.

16.1.4 Plan DXF: chainage ticks and a scale bar

The plan export adds two more layers alongside `ALIGNMENT` and `CONTOURS`:

- **CHAINAGE_TICKS** – short tick marks crossing the alignment perpendicular to its direction of travel, at a round chainage interval (again auto-chosen for roughly 8–20 ticks along the whole alignment), with a taller tick every 5th one.
- **SCALE_BAR** – a single reference line of an exact, round real-world length, with end ticks, placed clear of the drawing. Since the file is already true 1-unit-per-metre scale, measuring this one line with your CAD package's own measuring tool is an unambiguous sanity-check on scale.

16.1.5 File compatibility: why the DXF is written as R12 (AC1009)

The exported file declares itself as DXF version `AC1009` (DXF R12) in its header, and uses the classic three-part `POLYLINE`/`VERTEX`/`SEQEND` entity chain rather than the newer, simpler `LWPOLYLINE` entity. This is a deliberate compatibility choice, not an oversight: `LWPOLYLINE` needs at least DXF R14, and a file that declares a modern version such as `AC1015` (AutoCAD 2000) is expected by strict, ODA-based DXF readers – which is what Vectorworks and many other CAD packages use for DXF import – to also carry entity *handles*, a `BLOCK_RECORD` table and a populated `BLOCKS`/`OBJECTS` section. An earlier version of LineTerra's DXF export declared `AC1015` without any of that supporting structure; it imported without complaint into Mathematica (whose importer is much more lenient) but was rejected outright by Vectorworks with an ODA error. DXF R12 has no such requirements – just a `TABLES` section declaring the layers used, then `ENTITIES` – making it the most broadly compatible DXF variant available, at the cost of giving up the newer entity type.

17 Examples

The six worked examples below are drawn directly from the vendor's own example notebooks (`LineTerra 6 min Road.nb`, `LineTerra Route.nb`, `LineTerra Pad.nb`, `LineTerra Dam.nb` and `LineTerra Road LAZ example.nb`), which cover roads, a long downloaded-elevation corridor, a building pad, a dam, and LAZ point-cloud terrain respectively – deliberately spanning the "more than roads" scope described in §1.

17.1 How to start

After installation, evaluate once per session:

```
Needs["LineTerra`"]
```

Then:

```
LineTerra["Project name"]
```

This prints the toolbar described in §5 for a project called "Project name". From here: define a Terrain, open the Plan/Profile window and design the alignment and longitudinal (vertical) profile, apply section templates along the alignment, and calculate earthworks – the same sequence as the Quick Reference workflow in §20.

If you are working from the raw source file rather than an installed paclet, the equivalent opening sequence – taken directly from the vendor's own example notebooks – is:

```
SetDirectory[NotebookDirectory[]];
Get["LineTerra.wl"]
LineTerra["Road A"]
```

17.2 Worked example: designing a road in six minutes

This follows `LineTerra 6 min Road.nb`, the vendor's own quick-start walkthrough, almost step for step.

1. **Load.** Install `LineTerra.wl` in a folder, then in a notebook `Get["/path/to/LineTerra.wl"]` and evaluate `LineTerra["Road C"]` to print the main toolbar.
2. **Terrain.** Click New Project then Terrain. Click GeoTIFF, browse to the GeoTIFF file (only one file is needed here), and click Load.
3. **Preview.** After the file loads, choose the background type and click Show Preview – a satellite photo backdrop is used in the original walkthrough. To use your own aerial photo instead, use the Photo + Contours button; the photo should cover the same extent as the terrain.
4. **Plan options.** Click Plan/Profile and pick display options – the walkthrough uses the Streetmap background, 5 m contours and a 100 m grid.
5. **Alignment.** Click on the plan to place a series of IPs defining the road layout. Clicking an existing IP lets you give it a circular arc or Bezier curve; any IP can also be dragged to move it.

6. **Profile.** Click Extract Profile once the alignment looks right. Increase the vertical exaggeration (VE) to make the resulting vertical section more readable, then add VIPs by clicking on the profile; each VIP can be dragged, and a K-value on it adds a vertical curve.
7. **Templates.** Click Templates, then Load Library and select `LineTerra_templates.ltl` – the walkthrough uses the supplied "simple two-lane motorway" template. The links up to the Split Point are always present (their thickness is construction thickness, not counted in earthworks); after the Split Point the section differs between cut and fill – here, a drainage ditch then a 1:1.5 upward slope in cut, or a 1:3 downward slope in fill. Left and right can differ but are mirrored in this example. MaxW (maximum batter search) and Template Margin control, respectively, how far a batter may search for the ground and how much extra width is drawn outside the ground intersection point on a section.
8. **Sections.** Click Sections and assign templates to chainage ranges – the walkthrough puts a bridge template over the low ground and the motorway template elsewhere. Gaps between two assignments with the same number of links are interpolated automatically, which is how a road can widen for a bus stop or slow lane. A chainage can be selected and computed individually to spot-check a section.
9. **Earthwork.** Click Earthwork, choose a chainage range and section interval, and compute. This reports total cut and fill, with a table and charts of quantities, and updates the plan to show the road and its batters.
10. **Fine tuning.** Moving the alignment or profile is reflected immediately in the plan and earthwork charts, so the design can be tuned by eye to minimise cost.
11. **3D view.** A 3D view can be drawn once earthwork exists; because the road has not yet been baked, cut sections are hidden below the (opaque) terrain unless the X-ray option is used to render the terrain semi-transparent. Occasional spikes in the 3D view are usually LIDAR noise from vegetation and can be reduced with the Despike button, or ignored – they do not affect the reported quantities. If the 3D view is slow to draw, reduce the mesh size.
12. **Baking.** When the design is finalised, Bake Road writes it into the terrain; switching the plan to the aerial background afterwards shows the finished road in the satellite view.

17.3 Worked example: a road on LAZ/LAS point-cloud terrain

This follows `LineTerra Road LAZ example.nb`, which repeats the same overall design sequence with a point-cloud-derived terrain instead of a GeoTIFF.

1. **Load.** Evaluate `LineTerra["Example 2 Road A"]` to open the main toolbar, then New Project then Terrain.
2. **Terrain.** Click LAZ/LAS Point Cloud. A `.laz` file needs PDAL or LASzip on the system path (e.g. `brew install pdal` – see §6.2); a `.las` file is read directly with no external tool. Preview the terrain in the normal way once it loads.
3. **Alignment and profile.** Click Plan, add IPs to lay out the alignment, then Extract Profile as usual.
4. **Templates.** Open a template library (rather than building one from scratch) and pick the templates needed – the walkthrough uses the library's motorway and bridge templates together, the bridge spanning the low section.

5. **Sections and earthwork.** Assign the templates to the alignment, then compute Earthwork – the plan updates and the quantities are reported exactly as with any other terrain source. With Earthwork left on Auto, adjusting the profile or alignment updates the quantities and plan instantly.
6. **Bake and export.** The road can be baked into the terrain and a 3D view produced; the plan or the sections can then be exported to scale as usual.

The point that matters here is that once a LAZ/LAS terrain has been loaded and gridded, every downstream step – alignment, profile, templates, sections, earthwork, baking, export – is identical to the GeoTIFF workflow above. The terrain source only changes how the ground itself is obtained.

17.4 Worked example: an embankment dam

This follows `LineTerra Dam.nb` – a direct illustration of the "designing dams" use mentioned in §1, built with the ordinary road/template machinery rather than the polygon tool.

1. **Terrain and alignment.** Click Plan/Profile/EW, then Geomap and Update Terrain to get a contoured map with tooltips showing elevation. Click to place IP1 and IP2 across the valley to be dammed (precise coordinates can be typed into the IP Coordinates table instead of clicking).
2. **Profile.** Scroll below the plan and click Extract Profile.
3. **Template.** Open Templates, load the template library, and either draw a section from scratch or start from the supplied Embankment preset, adjusting the batters and crest width as needed.
4. **Assign and check.** In Sections, assign the Embankment template, choose a chainage, and click Compute to check the cross-section at that point.
5. **Earthwork.** Open Earthworks, choose a section interval, and compute. This reports the total fill required for the embankment, plus a small amount of cut wherever the profile cuts into the bank.
6. **Review.** The plan shows the embankment outline; switching to the aerial background with Update Terrain overlays it as a transparent layer on the satellite image. Moving the IPs or VIPs updates the plan, sections and quantities automatically.
7. **Bake.** Once the design is satisfactory, bake the dam into the terrain.

Further work. Filling the dam requires material from somewhere. Two approaches work within LineTerra: add a polygon (§7.1) as a quarry and adjust it until it yields enough material, or use the same alignment to cover both the quarry and the dam, with multiple templates along its length switching between quarrying and filling as needed.

17.5 Worked example: a building pad with the polygon tool

This follows `LineTerra Pad.nb`, using the polygon tool of §7.1 rather than an alignment, for a pad with vertical walls on one side and a 60° sloped wall on the other.

1. **Refine and bake a first pass.** With the polygon drawn and a Formation RL set, choose Refine → 1 m, click "← Mean ground", then Bake – establishing a baseline pad before fine-tuning its edges.

2. **Unbake and re-profile.** Click Unbake, then Extract Profile along the polygon's control line and add VIPs where the wall geometry needs to change.
3. **Define the wall sections.** Open Templates and create two: one where the first side of the polygon is vertical (Cut = False, near-zero width, a Wall-type link) and the remainder of the boundary slopes at 60°; and a second, purely sloped section at 60° for where no vertical wall is needed.
4. **Assign.** In Sections, assign the wall template to the vertical-face chainage range and the sloped template elsewhere around the polygon boundary.
5. **Earthwork.** Compute earthworks – the values shown in the Earthwork window itself can be ignored at this stage; what matters is the quantities reported after baking.
6. **Bake and check.** Bake the polygon. The reported quantities and the 3D view can then be reviewed; small black vertical artefact lines at the base of the walls in the 3D view are normal, caused by noise where the source GeoTIFF represents the ground surface including vegetation, and can be ignored.
7. **Make permanent (optional).** Once satisfied, make the bake permanent and update the terrain; sections can then be drawn through the slope, or further pads added alongside it, on the updated ground.

17.6 Worked example: a long corridor road from downloaded elevation data

This follows `LineTerra Route.nb`, a 160 km preliminary road design in Canada that uses the Corridor terrain source to keep a very long alignment's terrain data to a workable size.

1. **Load.** Evaluate `LineTerra["Route"]`, then New Project then Terrain.
2. **Corridor.** Click Route Corridor and enter the latitude/longitude of at least three points along the assumed route centreline (Add gives more rows as needed).
3. **Corridor parameters.** Set the half-width of the corridor, the spacing of interpolated elevation points along it, and the number of cross-samples within the half-width – an odd number, so one sample falls exactly on the centreline. (The Elev. sample field is currently unused.) LineTerra always downloads elevation at the maximum resolution available for the area and interpolates it to the requested spacing and sample count; coarse in absolute terms over 160 km, but sufficient for preliminary design.
4. **Preview.** Show Preview gives a plan with the corridor unrolled along the alignment in a horizontal (Eastings/Northings) direction.
5. **Plan and geographic view.** Pressing Plan and inserting IPs lays out the road along the unrolled corridor. A red marker on the plan flags a change in source-tile angle; Geographic View shows each terrain tile at its true angle so the cause is visible – a straight road crossing such a point will show a kink in plan, normally resolved by inserting a curve there.
6. **Zooming.** Parts of the route can be inspected in more detail by increasing the Plan width (with scroll bars appearing if it exceeds the window) or using the zoom button, with the aspect ratio adjusted as needed.
7. **Profile, templates, sections, earthwork.** From here the process is identical to any other terrain source: calculate the profile, assign section templates, and compute sections and earthwork

quantities.

18 Notes on Timing, Accuracy and Quantities

18.1 Timing

There are a few places where LineTerra takes a little while to complete its work – normally not enough to be a problem. Fetching geographic data depends on the internet connection. Previewing the terrain and drawing the plan may need a little patience, but should only take a minute or two. Fine-tuning the design and seeing the results in the earthwork quantities is close to instantaneous. Displaying the 3D view may take a while. The overall design process is nonetheless very fast – the vendor's own six-minute-road example (§17) took under 6 minutes end-to-end.

18.2 Accuracy

The main thing to be aware of is accuracy. You can, and should, check LineTerra's own calculation of quantities, but it cannot be more accurate than the data supplied to it:

- Where terrain comes from a GeoTIFF or a LAZ/LAS file, accuracy is that of the survey file itself.
- Where Wolfram's `GeoElevationData` is used, take care to ensure the resulting quantities and sections are suitable for the purpose at hand. This is normally sufficient for initial design, but that depends on the position and size of the project. A common approach is to use `GeoElevationData` for the preliminary design (as in the 160 km corridor example above), then use that to scope a detailed survey, and design in detail against the survey data once it's available.
- When using Lat/Lon for elevation data, LineTerra prints a message showing the actual spacing of elevation points (as an area per pixel) – it always requests the best data available for the location, but the underlying resolution still varies by where in the world the terrain is.
- Pads are normally used for smaller areas, so a Refine control is provided on the polygon/pad tool (available once the pad is baked). It interpolates the terrain to a finer grid, which is most useful for showing battered walls correctly and for avoiding the erroneous results a coarse mesh can otherwise produce, especially in the 3D view. If refining would produce more than 4 million pixels, it is skipped and a message suggests a coarser value instead.

18.3 Quantities

Quantities are calculated using average-end-area volumes between sections, with an eccentric end-area (curvature) correction. On a curve, the section footprint overlaps on the inside of the bend and gaps on the outside; a plain end-area × centreline-length calculation would double-count the inside. Each area is instead scaled by $(1 - e/R)$, where e is the centroid offset from the centreline (positive right) and R is the signed radius (positive curving right). Inside the bend, e and R share sign, so the factor is < 1 (the volume shrinks); outside, it's > 1 (the volume grows). The factor is clamped to $[0.2, 1.8]$ so that a wide section on a tight radius can't go negative or explode. Straight sections $(R = 0)$ get a factor of exactly 1.

19 Tips & Troubleshooting

- **A cross-section looks unexpectedly narrow.** Check, in the Templates panel, that the template assigned at that chainage actually has Right/Left links defined, and that a chainage-range assignment covers the section you're viewing – a chainage with no matching assignment falls back to the first template in the list, and a template with no links at all draws the batter point right at the centreline.

- **A cross- section erroneously shows the full batter width rather than stopping at the point where the batter hits the ground** This is most likely to be at a change from cut to fill and may be ignored. It will be affected by the spacing of the sections.

- **LAZ terrain won't load / "No LAZ decoder found".** See §6.2 – install PDAL or LASzip and, if it was installed via conda, set `$ltLAZDecoderPath` explicitly.

- **GeoMap/Aerial or Lat/Lon terrain won't load.** Both require an internet connection; Relief, Contour and Hybrid plan modes, and GeoTIFF/XYZ/LAZ/Corridor terrain, all work offline.

- **OSGB36 or an EPSG export fails with "No originLL".** That CRS needs a lat/lon terrain origin. Load terrain via the Lat/Lon source (or another georeferenced source that sets one), or use the Local grid option, which never needs one.

- **Bake Road is unavailable.** Compute Earthwork at least once for the active road first – baking reads the computed earthwork sections.

- **A National Grid conversion looks slightly off.** See the British National Grid adjustment note in §15 – check any national grid conversion against known values before relying on it.

- **Windows are piling up / Mathematica feels sluggish.** Use Close Views to close every LineTerra window for the session before loading a large new terrain or switching projects.

- **Calling an lt-prefixed helper directly fails after loading.** That's by design (§4) – lowercase-prefixed symbols are private implementation detail in `LineTerra‘Private‘`, not part of the public API. Use the matching capitalised LT-prefixed function instead, or open an issue if you have a genuine need to call the private helper directly.

- **A DXF export has no visible text/labels.** That's by design, not a bug – see §16.1. Identify geometry by its layer name, and use the SCALE/CHAINAGE_TICKS/ SCALE_BAR tick marks as a dimension reference instead.

- **A DXF export fails to import, or a CAD package reports an ODA/validation error.** Check you're using LineTerra v1.1 or later – an earlier revision declared a newer DXF version (AC1015) that strict, ODA-based readers rejected for missing supporting structure (entity handles, `BLOCK_RECORD`, `OBJECTS`); the file is now written as R12 (`AC1009`), which has no such requirements and is the most broadly compatible DXF variant available (§16.1).

20 Quick Reference

This mirrors the in-app Help panel (toolbar → Help).

20.1 Workflow

1. Click “Terrain” – load a GeoTIFF, fetch from lat/lon, load XYZ or LAZ/LAS, or define a corridor.
2. Click “Plan/Profile/EW” – opens the main design window.
3. In the plan: click to add IPs, drag to move them.
4. Click “Update Terrain” – renders the terrain background.
5. Use the profile panel below the plan to set VIPs and grades.
6. Click “Templates” – define cross-section templates (road, batters, ditches).
7. Click “Sections” – view cross-sections along the alignment.
8. Click “Earthwork” – compute cut/fill volumes. Move IPS or VIPs to refine quantities.
9. Click “Export” – export alignment, profile, earthwork, coordinates.
10. Click “Bake Road” – embeds the road into the terrain for multi-road projects or 3D views.

20.2 Multiple roads

```
LineTerra["Road A"] (* open or switch to Road A’s toolbar *)
LineTerra["Road B"] (* open Road B; Road A’s state is saved automatically *)
```

Use Intersect to find where the current road crosses another stored road, and Bake Road to embed a finished road into the terrain before designing the next one on top of it.

21 Beyond Roads

Although this program has been described in the sense of designing roads, with a little imagination, it can be used for many other purposes. For instance,

- laying out drainage networks; Use straight alignments between IPS/VIPS. Layout the main route first then use multiple "roads" and intersection points to form the sub branches. You can even incorporate balancing ponds or similar.
- laying out pipelines;
- calculating volumes eg. for dredging; Use the pad option to provide a volume down to a base for a terrain defined by the existing level survey. Then do the same for the proposed levels on a terrain file based on the proposed or post dredged survey.
- checking slope/cliff movement or stability; Run a set of cross sections along the cliff at any desired spacing for use in a slope stability program or compare slopes from two different surveys to check on movement.
- layout development particularly in sloping ground; use a "road" or pad to shape the overall site then use pads for individual building plots or access paths.

Whatever you use it for I hope you enjoy it and find it useful.

****