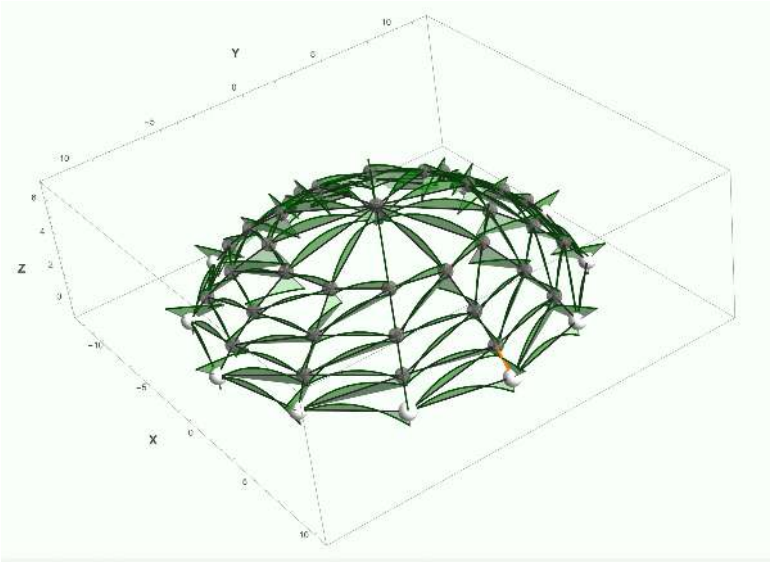
Framica

3D Space Frame Workbench

Version Beta 1.00

User Manual and Worked Examples



A Mathematica-based structural analysis workbench for 3D space frame modelling, FEA, and design.

Dr. Malcolm Woodruff PhD. M.I.C.E., M.I.Struct.E, F.G.S.

This document was generated from the Framica Beta 1.00 source code. Behaviour and interface elements reflect the implementation at the time of writing.

Contents

Part I

User Manual

Chapter 1

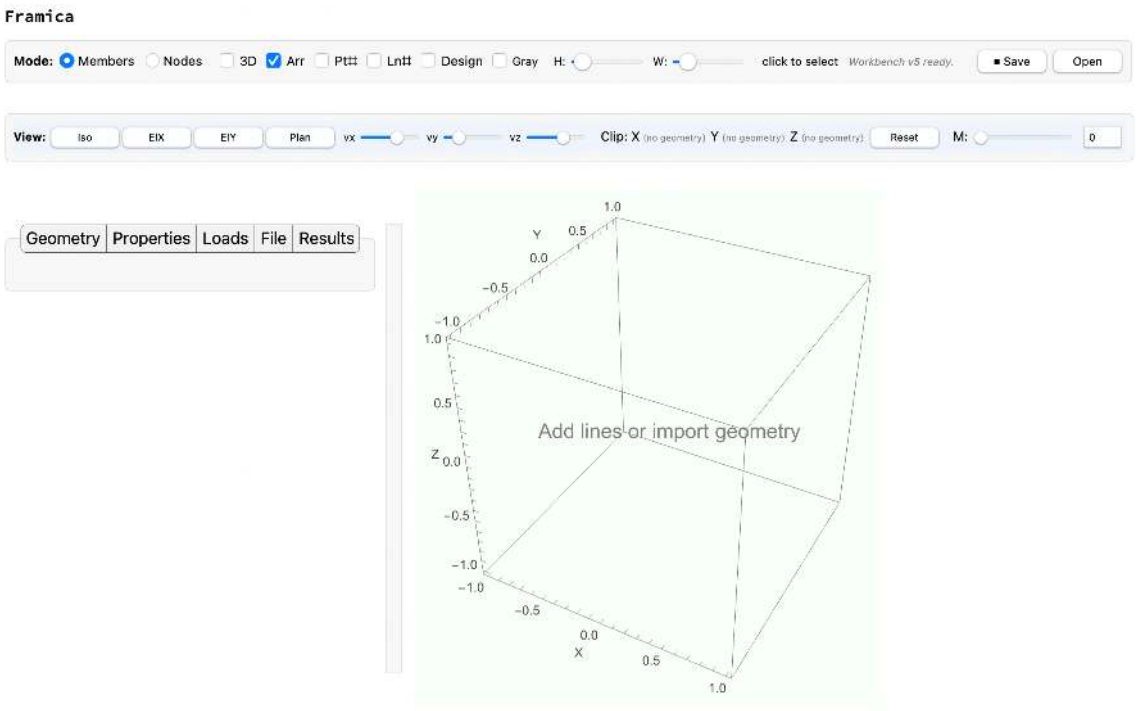
Overview

Framica is a 3D structural workbench implemented as a Mathematica `DynamicModule`. It provides an interactive environment for building, loading, and analysing space-frame models without leaving the Mathematica notebook. Its principal capabilities are:

- Interactive 3D geometry editing (members and nodes)
- Steel section assignment from European, UK, and American section libraries, plus user-defined custom sections
- Multiple load cases (Primary and Combination types)
- Linear-elastic FEA solver using full 3D Euler-Bernoulli beam elements
- Thermal loads: uniform temperature change (axial) and through-depth gradient (bending)
- Natural-frequency (modal) analysis with animated mode shapes
- Result diagrams: bending moments M_y and M_z, shear V_y and V_z, axial force N, torsion T_x, and deflected shape
- Member detail reports and a programmatically accessible member database
- Save/load project files (`.wl` format), CSV and DXF import/export
- A programmatic interface (`FramicaImport`, `FramicaModal`, named run ids) for scripting and geometry exchange

Chapter 2

Interface Layout



The workbench is arranged in two main zones displayed side by side:

Left panel A scrollable tab panel (approximately 310 px wide) containing five tabs: *Geometry*, *Properties*, *Loads*, *File*, and *Results*.

Right panel The 3D interactive viewport. Width and height are controlled by the **W** and **H** sliders in the top toolbar.

Above both zones sit two horizontal toolbars: the *mode/display toolbar* (topmost) and the *view/-clip/member toolbar* (second row).

Chapter 3

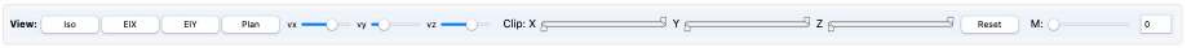
Toolbars

3.1 Top Toolbar

Table 3.1: Top toolbar controls

Control	Function
Mode: Members / Nodes	Toggle whether clicks in the viewport select members or nodes
3D checkbox	Render members as extruded 3D cross-sections (requires section assigned)
Arr checkbox	Show member direction arrows (start → end)
Pt# checkbox	Show node index labels
Ln# checkbox	Show member index labels
Design checkbox	Highlight and label design groups (continuous members across segments)
Gray checkbox	Gray out selected members so the model remains navigable during selection
H / W sliders	Resize viewport height and width
Selection display	Shows currently selected members (M: . . .) or nodes (N: . . .)
Debug message	Status and feedback from the most recent action
Project filename	Displayed in blue italic when a file is open
Save button	Quick-save to the current project file; opens a file dialog on first save
Open button	Load a previously saved `.wl` project file

3.2 Bottom Toolbar — View Controls



The second toolbar row controls the 3D camera and spatial clip box.

Table 3.2: Preset view buttons

Button	**Direction**
Iso	Isometric (default)
ElX	Elevation along the X axis
ElY	Elevation along the Y axis
Plan	Plan view (from above)

3.2.1 Preset views

3.2.2 Fine view control

Three sliders (**vx**, **vy**, **vz**) set a custom `ViewPoint` vector, ranging from -4 to $+4$ (vz from -2 to $+4$).

3.2.3 Clip box

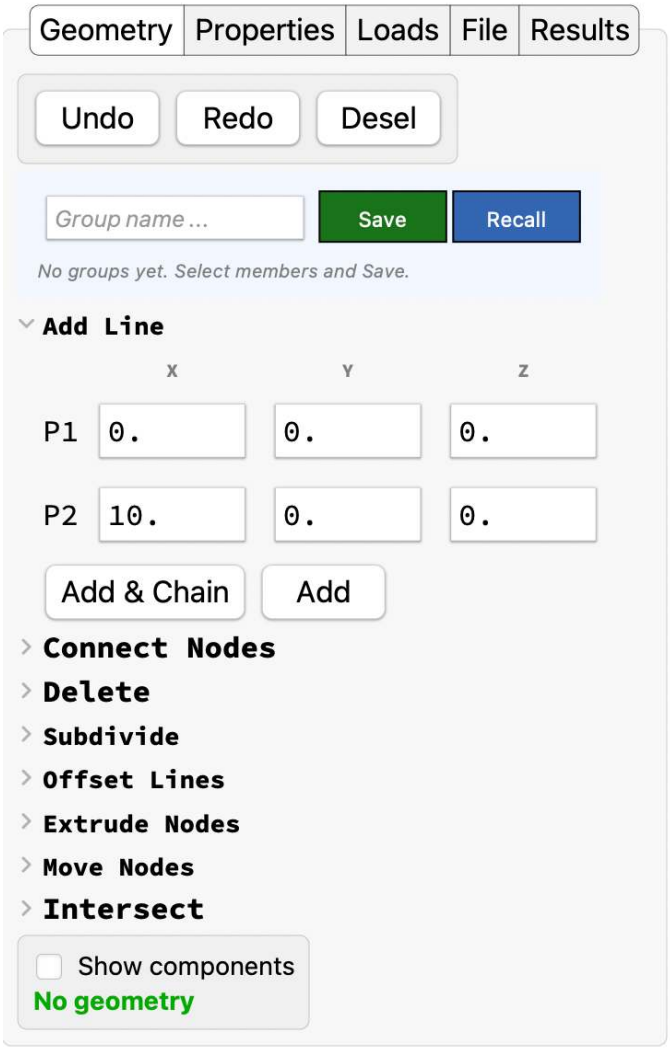
Three `IntervalSlider` controls (X, Y, Z) restrict which part of the model is visible. The **Reset** button restores the clip box to the full model extent plus a 1 m margin.

3.2.4 Member selector

The **M:** slider and adjacent numeric input field step through members by index. Setting the value to 0 clears the selection.

Chapter 4

Geometry Tab



All geometry operations are grouped in collapsible `OpenerView` panels. The general workflow is: add lines, then connect, split, transform, or extrude nodes.

4.1 Undo / Redo / Deselect

Every geometry and property action pushes a full model snapshot to the undo stack. The stack is unlimited within a session.

Table 4.1: Undo toolbar buttons

Button	Action
Undo	Revert the last action
Redo	Reapply the last undone action
Desel	Clear all selected members and nodes
Sel All	Select all members (or all nodes in Nodes edit mode)

4.2 Add Line

Enter start point **P1** and end point **P2** as X, Y, Z coordinates:

- **Add & Chain** — adds the segment and moves P1 to P2 so the next line continues from that point.
- **Add** — adds the segment without chaining.

The model is automatically cleaned (duplicate lines merged, collinear points resolved) after each addition.

4.3 Connect Nodes

Switch to **Nodes mode**, select exactly two nodes in the viewport, then click **Apply**. A new member is created between them.

4.4 Delete

- **Lines** — deletes all selected members.
- **Points** — deletes all selected nodes and any members connected to them.

Member properties and load assignments are preserved on surviving members.

4.5 Subdivide

Splits selected members into smaller segments.

Table 4.2: Subdivide modes

Mode	Input	Description
Number	$n \geq 2$	Divide into n equal segments
Ratio	Comma-separated values, e.g. `1,2,1`	Divide proportionally
Length	Comma-separated lengths (m)	Fixed-length segments; remainder auto-added if sum $<$ member length

Distributed loads (UDL, Trapezoid) on subdivided members are automatically redistributed to the new segments.

4.6 Offset Lines

Copies selected members translated by a vector $(\Delta x, \Delta y, \Delta z)$.

- **Sections** checkbox — new members inherit the section of the originals.
- **Loads** checkbox — member loads are copied to new members in all Primary load cases.

The original members remain selected; clicking **Apply** again repeats the offset.

4.7 Extrude Nodes

Extends selected nodes by a vector, creating a new member between each original node and its extruded copy.

4.8 Move Nodes

Translates selected nodes in-place. All connected members move with them; member properties are preserved.

4.9 Intersect

Finds and splits members that cross each other without a shared node.

Table 4.3: Intersect modes

Mode	Behaviour
Off	No intersection performed
Selected	Intersects only selected members against the full model
All	Intersects the entire model

4.10 Connectivity Checker

The **Show components** checkbox colours members by connected component. A fully connected model shows **Connected**; multiple disconnected sub-frames are flagged in orange with the component count.

Chapter 5

Properties Tab

5.1 Section Assignment

1. Choose a **Library**: European, UK, American, or any custom library.
2. For standard libraries, choose a **Type** (e.g. IPE, HEA, UB, UC, CHS...).
3. Choose a **Section** from the dropdown.
4. A live **preview** shows a 2D profile sketch and a scrollable table of section properties with units.
5. Select members in the viewport, then click **Assign**.

The **Custom section editor** button opens an external editor for user-defined sections (see Chapter 13).

5.2 Orientation

Sets the cross-section rotation about the member axis, in degrees. Select one or more members, enter the angle, and click **Assign**. When a single member is selected, the field auto-populates with its current orientation.

5.3 Member Releases

Table 5.1: Member release types

Option	Meaning
Fixed	Full moment continuity (default)
Pinned	Releases all moments at that end
Pin My	Releases M_y only
Pin Mz	Releases M_z only

Set start and end releases independently, then click **Assign**.

5.4 Force Type

5.5 Node Supports

Select nodes, choose a condition from the dropdown, then click **Assign**.

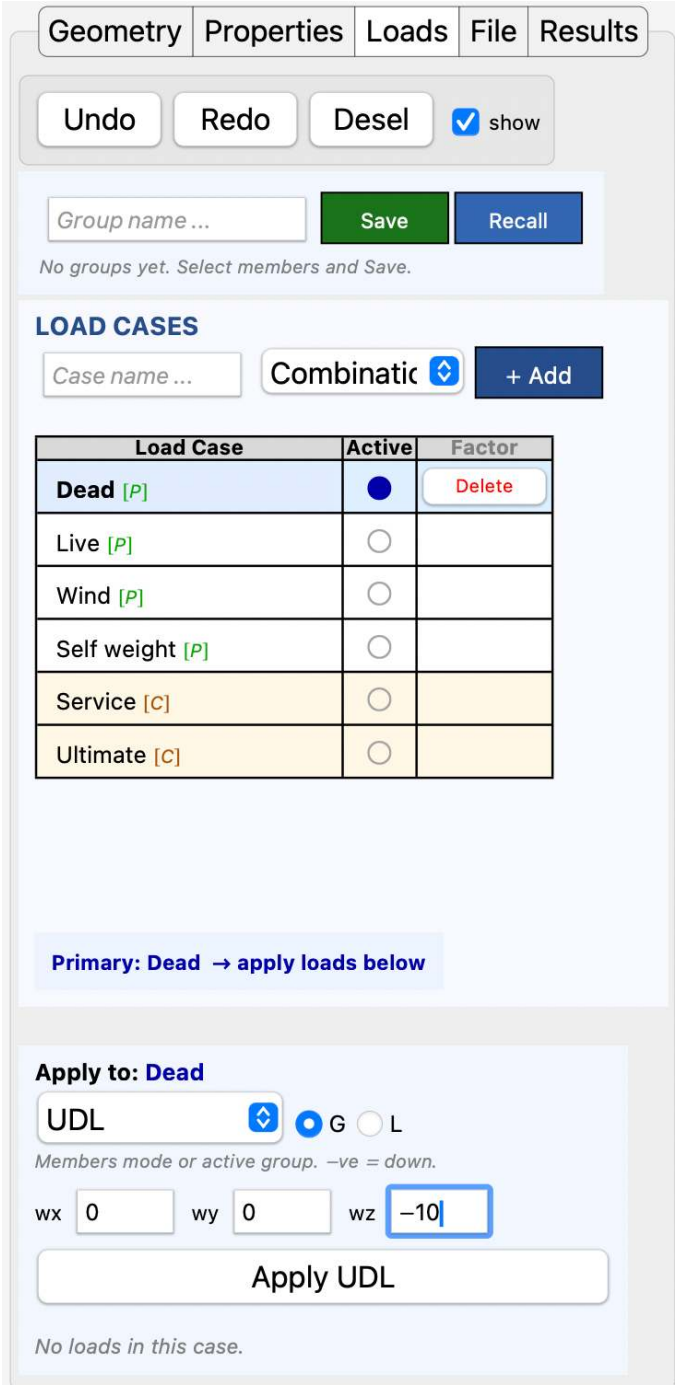


Table 5.2: Member force types

Option	Behaviour
Tension+Compression	Standard beam/column element (default)
Tension Only	Member goes slack under compression — used for cables and cross-bracing diagonals
Compression Only	Member goes slack under tension

Table 5.3: Node support types

Type	DOFs fixed
Fixed	All 6 DOFs (3 translations + 3 rotations)
Pinned	All 3 translations; rotations free
Fixed (Internal)	No support (default for unsupported nodes)

5.6 Spring Supports

Adds translational spring stiffness (kN/mm) to selected nodes. Enter K_x, K_y, K_z (set to 0 to omit a direction), choose the spring type (**Both**, **Tension only**, or **Compression only**), and click **Assign**. **Clear springs** removes spring data from selected nodes.

5.7 Model Summary

A panel at the bottom of the Properties tab shows member count, node count, and total mass (kg) based on assigned sections.

Chapter 6

Loads Tab

6.1 Load Cases

Create and manage load cases at the top of the Loads tab.

- **Primary** cases hold direct load entries (forces, moments, UDLs, etc.).
- **Combination** cases are weighted sums of Primary cases. Enter a factor next to each Primary case name; zero or blank excludes that case.

Click the circle button next to any case name to make it the active case.

6.2 Load Types

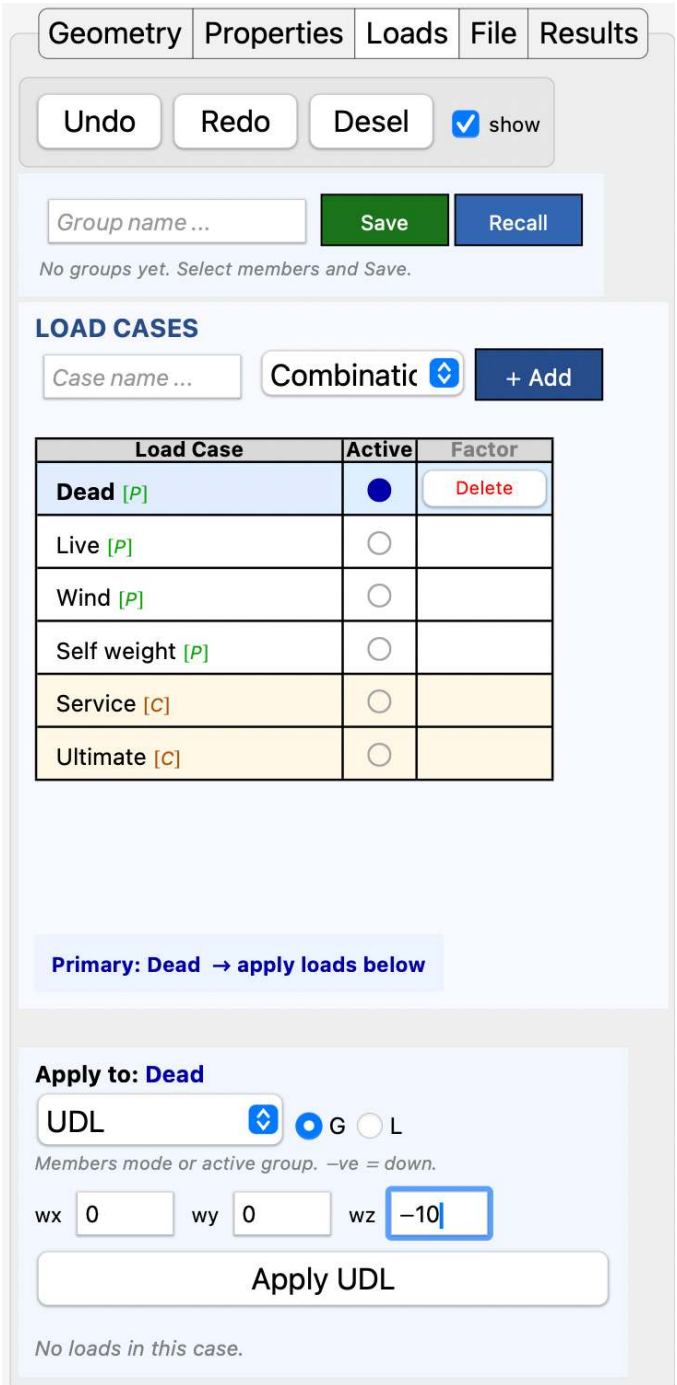
Table 6.1: Available load types

Type	Applies to	Inputs
Point Force	Selected nodes	F_x, F_y, F_z (kN)
Point Moment	Selected nodes	M_x, M_y, M_z (kNm)
UDL	Selected members or active group	w_x, w_y, w_z (kN/m). Negative = downward in Global Z
Trapezoid	Selected members or active group	w_{start}, w_{end} (kN/m); from, to (m); direction d_x, d_y, d_z
Member Point	Selected members or active group	Position (fraction 0–1 or metres from start); F_x, F_y, F_z (kN)
Gravity	All sectioned members automatically	g (m/s^2) — applies self-weight as a UDL
Thermal	Selected members (*all* if none selected)	Uniform ΔT (°C); Gradient ΔT (°C) top−bottom

Design Members mode: When the **Design** checkbox is active, UDL, Trapezoid, and Member Point loads are applied across the entire continuous design group, with position interpreted relative to the total group length.

6.3 Thermal Loads

A **Thermal** load applies a temperature change to members. Two independent effects are available:


Geometry
Properties
Loads
File
Results
Undo
Redo
Desel
show
Group name ...
Save
Recall
No groups yet. Select members and Save.
LOAD CASES
Case name ...
Combinatic
+ Add
Load Case
Active
Factor
Dead [P]
Delete
Live [P]
Wind [P]
Self weight [P]
Service [C]
Ultimate [C]
Primary: Dead → apply loads below
Apply to: Dead
UDL
G
L
Members mode or active group. −ve = down.
wx
0
wy
0
wz
−10
Apply UDL
No loads in this case.

- **Uniform** ΔT — a uniform temperature change of the member. It produces an axial thermal force $N = EA\,\alpha\,\Delta T$ acting along the member's *local* axis (so it is correct for members at any orientation). A restrained member develops this force as internal stress; an unrestrained member simply expands by $\alpha\,\Delta T\,L$.
- **Gradient** ΔT — a through-depth temperature difference (top minus bottom, over the section depth h). It produces thermal curvature and bending moments $M = EI_z\,\alpha\,\Delta T_{\text{grad}}/h$ about the local z (major) axis.

The coefficient of thermal expansion α defaults to steel ($12 \times 10^{-6}/°\text{C}$) and may be overridden per section by adding an `"Alpha"` field to the section data.

Thermal loads apply to the currently selected members, or — if nothing is selected — to *all* members (as Gravity does). Thermal loads are ordinary load-case entries, so they combine with forces, UDLs, etc. and appear in the FEA results. A pure uniform ΔT moves members axially, which shows in the deformed shape and the total-displacement readout rather than in the transverse bending diagrams.

6.4 Load List

Below the **Apply** button a scrollable list shows all loads in the active case. Use **Debug last** to print the most recent load entry, **Clear all** to remove all loads from the active case, and **Del selection** to remove loads on currently selected nodes or members. The × button on each row removes that individual load.

Chapter 7

File Tab

Table 7.1: File operations

Button	Function
Save As .wl	Save the full project (geometry, properties, loads, UI state)
Load .wl	Open a previously saved project
Import CSV	Import geometry from two CSV files (nodes + elements). Prompts for each file sequentially
Export CSV	Export node table (`#,X,Y,Z,Release`) and member table (`#,N1,N2,...`) as two CSV files. These round-trip back through **Import CSV**
Import DXF	Import geometry from a `.dxf` file. Reads `LINE`, `LWPOLYLINE` and `POLYLINE` entities as members; Framica auto-numbers the nodes and members and assigns default properties
Export DXF	Export the current members as `LINE` entities to a `.dxf` file (geometry only) for exchange with CAD tools
Export to Print	Dumps the full model data structure to the Mathematica output cell

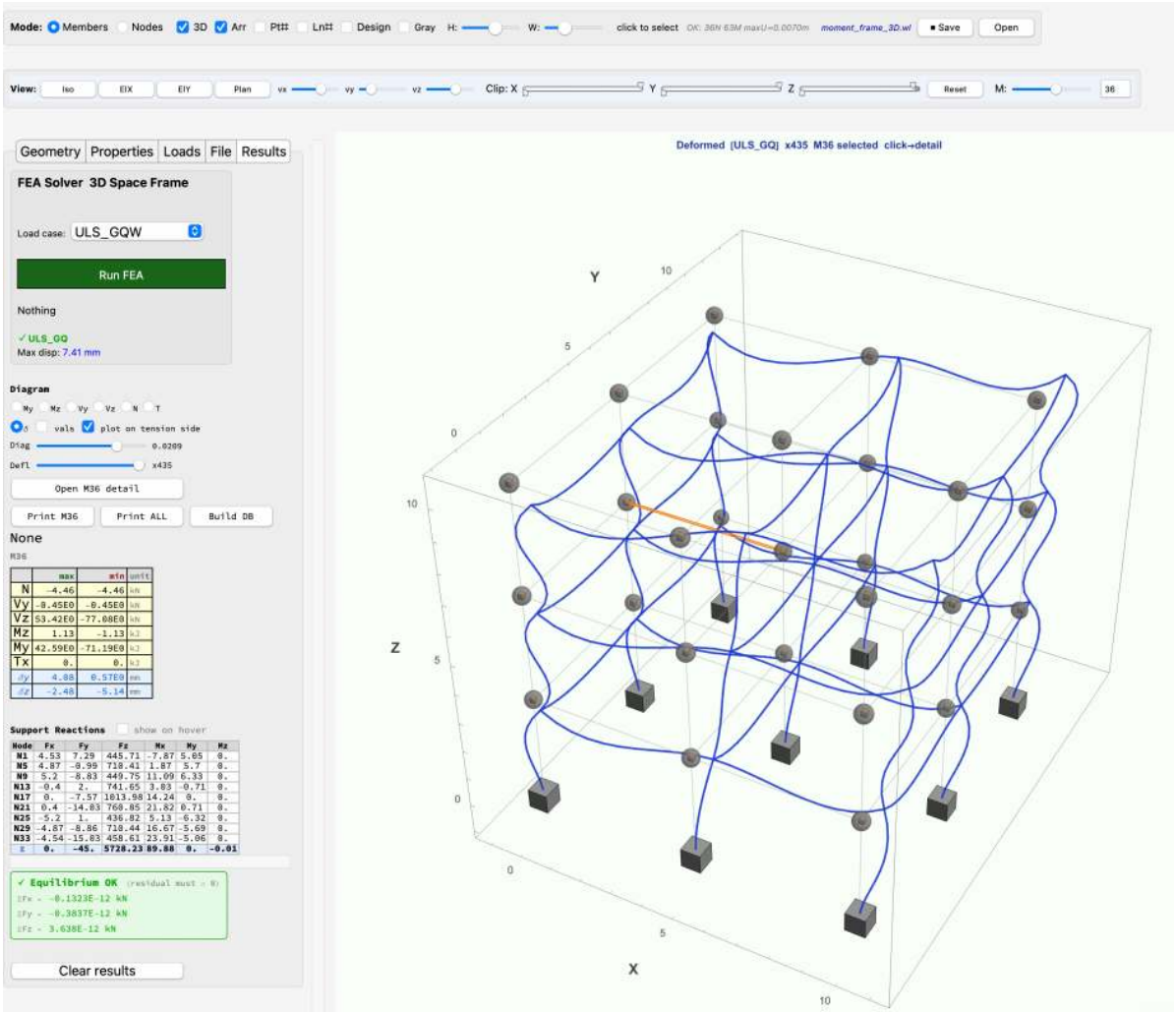
CSV format. The nodes file has columns `#, X, Y, Z, Release`; the members file has `#, N1, N2` followed by member properties, where `N1`/`N2` are node numbers. To import, choose the nodes file first, then the members file.

Chapter 8

Results Tab

8.1 Running FEA

1. Select a load case from the dropdown (Primary or Combination).
2. Click **Run FEA** (green button).
3. On success the viewport switches to Results mode. On failure a red error panel lists common causes.



8.2 Diagram Types

Table 8.1: Result diagram types

Button	Diagram
My	Bending moment about local y (major axis)
Mz	Bending moment about local z (minor axis)
Vy	Shear force in local y
Vz	Shear force in local z
N	Axial force
T	Torsion T_x
δ	Deformed shape

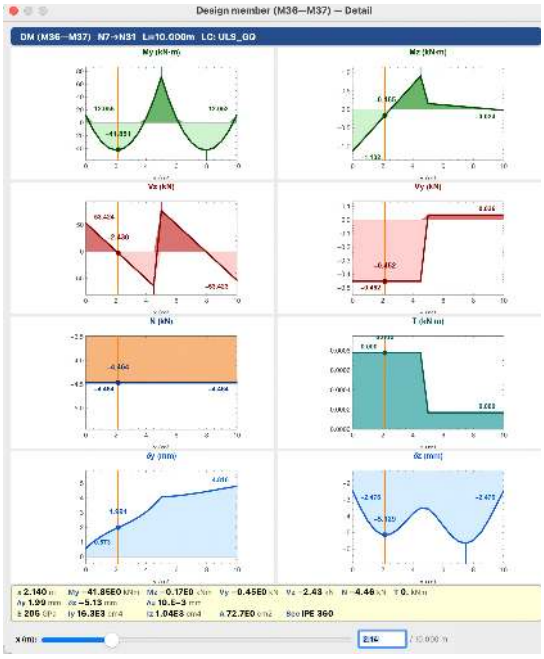
The **Vals** checkbox overlays numerical values on the diagram. **Plot on tension side** draws moment diagrams on the tension face of the member (common in UK practice).

8.3 Scale Controls

Table 8.2: Diagram scale sliders

Slider	Controls
Diag	Diagram scale (logarithmic, 10^{-6} to 0.5 m/kN)
Defl	Deflection amplification factor (0.5× to 500×)

8.4 Member Detail Report



- **Open Mn detail** — opens an interactive panel inside the workbench.
- **Print Mn** — creates a separate notebook and sends to the printer.

- **Print ALL** — prints all members (or design groups) as one notebook.

8.5 Build DB

Compiles all member results into a Mathematica `Dataset` stored in `$fwMemberDBAll` and `$fwMemberDB`. The dataset is also printed to the output cell, enabling downstream programmatic access to forces, sections, and design-group data.

8.6 Member Force Summary

A table shows max and min values for N, V_y, V_z, M_z, M_y, T_x (in kN / kNm) and deflections δ_y, δ_z (in mm) for the selected member, or all members if none is selected.

8.7 Support Reactions

A table lists reactions F_x, F_y, F_z (kN) and moments M_x, M_y, M_z (kNm) at every support node, with a sum row Σ. An equilibrium check panel confirms whether $|\Sigma F_x|$, $|\Sigma F_y|$, $|\Sigma F_z| < 0.05\,\text{kN}$. Green = equilibrium satisfied; red = potential model error.

8.8 Natural Frequencies (Modal Analysis)

The **Modal** panel computes the structure's natural frequencies and mode shapes. It assembles a consistent mass matrix alongside the stiffness matrix and solves the generalised eigenvalue problem

$$\mathbf{K}\,\boldsymbol{\phi} = \omega^2\,\mathbf{M}\,\boldsymbol{\phi},$$

reporting the lowest modes as frequencies $f = \omega/2\pi$ (Hz) and periods $T = 1/f$ (s).

1. Set the number of **Modes** to compute (default 6).
2. Click **Run Modal**. A table lists mode number, frequency (Hz) and period (s).
3. Click **Open mode-shape viewer** to open a separate, resizable window showing the animated mode shapes, with a mode selector, amplitude scale, and a play control.

- **Mass** for each member comes from the section's tabulated mass per metre (kg/m); bare members fall back to ρA with steel density $\rho = 7850\,\text{kg/m}^3$.
- **Supports** must be defined — modal analysis uses the same restraints and grounded springs as the static solver.
- Member end-releases are treated as rigid for modal analysis; the result is exact for rigidly-jointed frames.

Modal analysis is also available programmatically as `FramicaModal["id"]` — see the Programmatic Interface chapter.

Chapter 9

Viewport (3D Canvas)

9.1 Selecting Members and Nodes

- In **Members mode**: click any member line or extruded body to add it to the selection. Click again to deselect.
- In **Nodes mode**: click any node sphere to select it.

9.2 Viewport Indicators

Table 9.1: Visual symbols in the viewport

Symbol	**Meaning**
Triangle / pyramid (green)	Fixed support
Circle (green)	Pinned support
Coil / helix	Spring support
Arrow on member	Member direction (start → end)
Red highlight	Selected member
Dashed coloured line	Design group extent
Dn label	Design group index

Chapter 10

Groups Panel

The Groups panel appears at the top of the Geometry, Properties, and Loads tabs.

10.1 Saving a Group

1. Select members in the viewport.
2. Type a name in the **Group name** field.
3. Click **Save**.

Groups store member geometry (not indices), so they survive reindexing.

10.2 Group Table

Table 10.1: Group table columns

Column	Meaning
Visible	Show/hide group members in the viewport
Active	Include group in load application and **Recall**
Name	Group name (bold if active)
Members	Number of members in the group
×	Delete the group

Clicking **Recall** selects all members belonging to currently Active groups. When groups are Active, UDL, Trapezoid, Member Point, and Thermal loads are applied to all members in those groups without requiring manual selection.

Chapter 11

Programmatic Interface

Beyond the interactive workbench, Framica exposes a small set of public functions for scripting and for exchanging geometry with other Wolfram Language code.

11.1 Loading Framica

After installing the paclet, load it once per session before use:

```
Needs["MalcolmWoodruff`Framica`"]
```

Then open the workbench with `Framica[]`. Loading the package first ensures the symbol resolves to the paclet (rather than an empty global) and that the drawing and solver functions are available.

11.2 Run IDs — Multiple Projects in One Notebook

Calling `Framica["id"]` with a string tag opens a workbench bound to a named *run*. Each run keeps its own geometry, properties, loads and results, so several projects can live in one notebook without interfering:

```
Framica["Bridge"]
Framica["Tower"]
```

Re-evaluating a `Framica["id"]` cell restores that project rather than resetting it. Plain `Framica[]` (no id) remains a scratch workbench. Run state persists for the kernel session; use the **Save** button to keep a project on disk beyond a kernel restart.

Helpers:

- `FramicaRuns[]` — list the ids of all runs held this session.
- `FramicaRuns["id"]` — return the stored state of a run.
- `FramicaReset["id"]` — clear a run; `FramicaReset[]` clears all.

11.3 Importing Geometry

`FramicaImport["id", nodes, members]` loads externally-computed geometry (for example generated in the Wolfram Language) into a run:

- `nodes` — `{{nodeId, x, y, z}, ...}` or an association `<|nodeId -> {x, y, z}|>`.
- `members` — `{{n1, n2}, ...}` or `{{memberId, n1, n2}, ...}` of node-id pairs.

Node ids are used only to resolve connectivity; Framica renumbers nodes internally by position. After importing, open the run with `Framica["id"]`. For example:

```
FramicaImport["frame1",
{{1,0,0,0},{2,4,0,0},{3,4,0,3},{4,0,0,3}},
{{1,2},{2,3},{3,4},{4,1},{1,3}}];
Framica["frame1"]
```

11.4 Modal Analysis

`FramicaModal["id"]` runs natural-frequency analysis on a run's geometry and returns an association with `"Frequencies"` (Hz), `"Periods"` (s) and `"Modes"` (mode shapes). The option `"Modes"->n` sets how many modes to return (default 6). Supports must be defined on the run.

11.5 Documentation and Examples

- `FramicaManual[]` — open this user manual.
- `FramicaManual["Developer"]` — open the developer manual.
- `FramicaManual["CustomSections"]` — open the Custom Section Editor guide.
- `FramicaExamples[]` — open the folder of bundled example projects (load them from the workbench with **Load .wl**).

Chapter 12

Workflow Tips

1. **Typical modelling sequence:** geometry → section assignment → supports → load cases → Run FEA → inspect results → Build DB.
2. **Save often.** The **Save** button quick-saves; the project filename is shown in blue-italic when a file is open.
3. **Design Members** should be enabled when your frame contains continuous beams modelled as multiple collinear segments.
4. **Spring supports** are useful for flexible foundations. Set stiffness to 0 in any direction to omit a spring in that direction.
5. **Combination cases:** Create all Primary cases first, then add a Combination and enter factors. Run FEA on the Combination to see the combined response.

Chapter 13

Custom Section Editor

13.1 Opening the Editor

In the **Properties** tab, click the **Custom section editor...** button at the top of the Section panel. A floating dialog opens titled *Custom Sections*. It remains open while you continue working in Framica.

13.2 Step 1 — Choose or Create a Library

A **library** is a named collection of custom sections saved as a single `.wl` file on disk. All custom libraries appear alongside the built-in UK, European and American libraries in the section assignment popup in the Properties tab.

Selecting an existing library: Use the popup menu at the top of the editor.
Creating a new library: Select — *new library* — from the popup, type a name in the field that appears, then click **Create**. The library file is created immediately.

The status line below the popup confirms the library name, how many sections it contains, and the full path to the `.wl` file on disk.

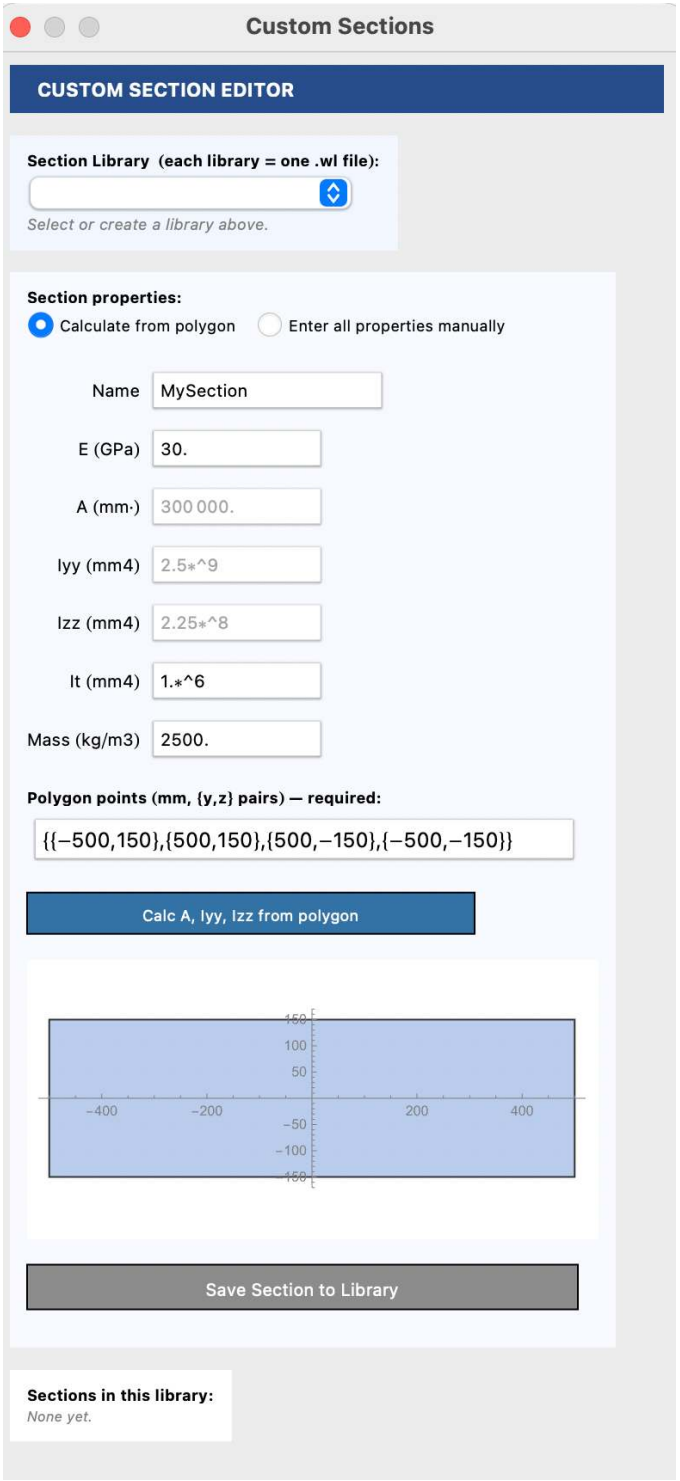
13.3 Step 2 — Choose a Calculation Mode

Two radio buttons control how section properties are derived:

Mode	Description
Calculate from polygon	You define the cross-section outline as a polygon. Area A, second moments I_{yy} and I_{zz} are calculated automatically when you press **Calc**.
Enter all properties manually	You type every property value yourself. The polygon field is optional and used only for the shape preview.

13.4 Step 3 — Enter Section Properties

Fill in the fields in the **Section properties** panel:


Custom Sections
CUSTOM SECTION EDITOR
Section Library (each library = one .wl file):
Select or create a library above.
Section properties:
Calculate from polygon
Enter all properties manually
Name
MySection
E (GPa)
30.
A (mm·)
300 000.
Iyy (mm4)
2.5*^9
Izz (mm4)
2.25*^8
It (mm4)
1.*^6
Mass (kg/m3)
2500.
Polygon points (mm, {y,z} pairs) — required:
{{−500,150},{500,150},{500,−150},{−500,−150}}
Calc A, Iyy, Izz from polygon
150
100
50
−400
−200
200
400
−50
−100
−150
Save Section to Library
Sections in this library:
None yet.

Field	Units	Notes
Name	—	Any text label, e.g. `300 conc slab` or `800 dia pile`
E	GPa	Elastic modulus. Steel ≈ 210 GPa, concrete ≈ 30 GPa
A	mm^2	Cross-sectional area. Greyed out in polygon mode until you press **Calc**
I_{yy}	mm^4	Second moment about local y-axis. Greyed out in polygon mode
I_{zz}	mm^4	Second moment about local z-axis. Greyed out in polygon mode
I_t	mm^4	Torsional constant. Always entered manually in the first instance; computed in the background after saving when a polygon is available
Mass	kg/m^3	Material density, used for self-weight calculation

13.5 Step 4 — Define the Polygon

The polygon is the cross-section outline in the **local** y–z **plane**, specified as a list of $\{y, z\}$ coordinate pairs in **millimetres**. They should follow the exterior of the polygon in a clockwise direction. It is not necessary to close the polygon as the last link will be assumed automatically.

Note: The Mathematica function `CirclePoints` enables easy calculation of vertices for regular polygons.

The polygon is used to:

- calculate A, I_{yy}, I_{zz} automatically (polygon mode);
- compute the torsional constant I_t in the background after saving;
- draw a shape preview when the section is selected in the Properties tab.

13.5.1 Polygon Syntax

```
(* Rectangular 300 mm wide x 500 mm deep *)
{{-150, 250}, {150, 250}, {150, -250}, {-150, -250}}

(* Hollow section: outer contour then inner contour *)
{{{-200,100},{200,100},{200,-100},{-200,-100}},
 {{-150,75},{150,75},{150,-75},{-150,-75}}}
```

The editor supports both **solid polygons** (a flat list of $\{y, z\}$ pairs) and **multi-polygon sections** (a list of polygon lists), useful for hollow sections or compound shapes. A live preview appears below the input field as you type.

13.5.2 Calculating Properties from the Polygon

After entering the polygon, click **Calc A, Iyy, Izz from polygon**. The calculated values populate the greyed-out fields. You can then adjust them manually if required before saving.

The polygon coordinates define the cross-section shape in the member's local y–z plane. The local x-axis is along the member axis. Centre the polygon at the centroid for consistent behaviour with the standard section library.

13.6 Step 5 — Save the Section

Click **Save Section to Library**. The following happens:

1. The section is immediately added to the selected library in memory and written to the `.wl` file on disk.
2. The section becomes available in the Framica section popup straight away — no restart is needed.
3. A background computation starts to calculate the **torsional constant** I_t using a finite element Laplace solver on the polygon outline. This may take a few seconds to a minute depending on section complexity. When it completes, I_t is updated in the library file automatically.

The status line below the button confirms the save with a green tick, or shows an error message in red.

Torsional constant pending: If you close Mathematica before the background I_t calculation finishes, the section will have $I_t = 0$. Simply re-save the section to trigger the calculation again.

13.7 Managing Existing Sections

The bottom panel of the editor shows all sections in the selected library.

Select a section
Click a section name to highlight it. The name turns blue and the Delete button activates.

Delete a section
With a section selected, click **Delete [section name]**. The section is removed from memory and the `.wl` file is updated immediately.

Delete an entire library
Click **Delete entire library '[name]'**. A confirmation step appears before the library is removed and its `.wl` file deleted from disk.

13.8 Unit Conversions

The editor accepts inputs in everyday engineering units and converts them internally before storing:

Property	You enter	Stored as	Conversion
E	GPa	MPa	$\times 1000$
A	mm^2	m^2	$\div 10^6$
I_{yy}, I_{zz}, I_t	mm^4	m^4	$\div 10^8$
Mass	kg/m^3	kg/m	$\times (A \div 10^6)$
Polygon	mm	m	$\div 1000$

You never see these conversions — simply enter values in the units shown in the field labels.

13.9 Notes on the E Value for Custom Sections

The E value stored with each custom section **takes precedence** over the global `$fwEMod` setting used for the standard libraries. This means:

- Changing `$fwESteel` or `$fwEMod["UK"]` does *not* affect sections in custom libraries.
- To change E for a custom section, open the editor, delete the existing entry, and re-save it with the corrected E value.

13.10 Multi-Polygon (Hollow) Sections

Framica detects nested polygon lists automatically and treats them as composite or hollow cross-sections. The area and second moments are computed for the combined shape, which may describe back-to-back tees separated by a gap, a U-shaped concrete section, or a hollow steel or concrete section.

```
(* Hollow rectangular box: 400 x 200 mm, 10 mm wall *)
{{{-200,100},{200,100},{200,-100},{-200,-100}},
 {{-190,90},{190,90},{190,-90},{-190,-90}}}

(* I-section approximated by three rectangles *)
{{{-100,200},{100,200},{100,185},{-100,185}},
 {{-10,185},{10,185},{10,-185},{-10,-185}},
 {{-100,-185},{100,-185},{100,-200},{-100,-200}}}
```

For hollow sections, list the **outer contour first**, then the inner contour(s). The editor uses the sign of the enclosed area to determine which polygons are voids.

13.11 Workflow Summary

1. Open editor via **Properties** tab → **Custom section editor...**
2. Select an existing library, or create a new one.
3. Choose **polygon** or **manual** mode.
4. Enter **Name**, **E**, **It**, **Mass**.
5. *Polygon mode:* enter polygon coordinates and press **Calc A, Iyy, Izz from polygon**. *Manual mode:* enter **A**, **Iyy**, **Izz** directly.
6. Click **Save Section to Library**.
7. Close the editor (or leave it open).
8. In the **Properties** tab, select your custom library from the **Lib** popup.
9. Select members in the viewport and click **Assign**.

Part II

Worked Examples

Running the Worked Examples

There are two ways to work through the examples that follow: load a ready-made project file, or build the model yourself from the geometry-generator code.

Route A — Load a ready-made project (quickest)

Three of the examples are bundled with the paclet as complete projects (geometry, sections, supports and load cases): *portal_ shed_ 3D* (Example 1, Part C), *moment_ frame_ 3D* (Example 2) and *schwedler_ dome* (Example 3).

1. Load the paclet and copy the examples to a writable folder:
   ```
   Needs["MalcolmWoodruff`Framica`"]
   FramicaExamples[]
   ```
   `FramicaExamples[]` copies the bundled projects into a **Framica Examples** folder in your **Documents** directory and opens that folder. (The paclet's own copy lives in a read-only system location, so the files are copied out to somewhere you can browse to.)
2. Open the workbench:
   ```
   Framica[]
   ```
3. In the *File* tab, click **Load .wl** and choose a project from the **Framica Examples** folder (or first copy it to your Desktop and load it from there).
4. Go to the *Results* tab, pick a load case from the dropdown, and click **Run FEA**. Try **Run Modal** for natural frequencies.

Route B — Build it from the generator code

Each example includes a *geometry generator* listing that constructs the model in code and calls `fwSaveProject[...]` to write a `.wl` file (to your Desktop by default). The generator uses Framica's internal functions, so **open the workbench once first**:

```
Needs["MalcolmWoodruff`Framica`"]
Framica[]
```

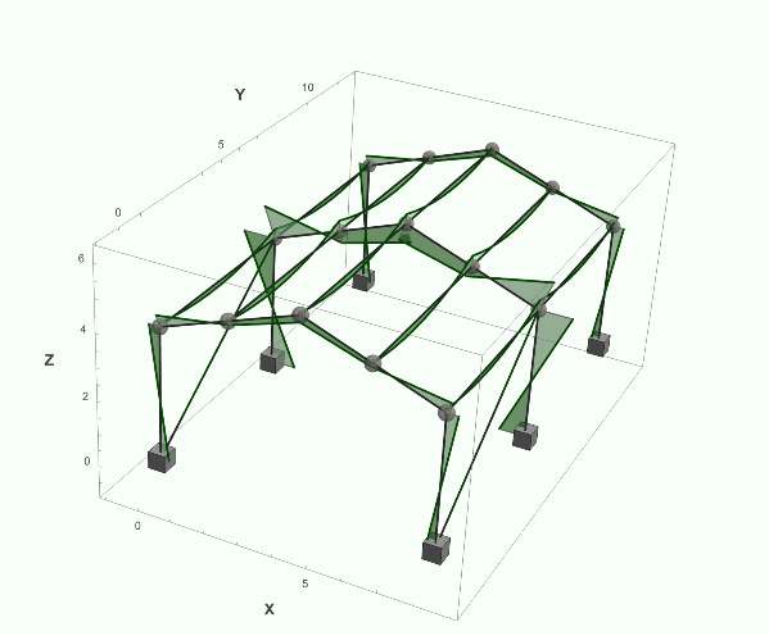
Opening `Framica[]` puts the internal context (`MalcolmWoodruff`Framica`Private`) on `$ContextPath`, which makes `fwSaveProject`, `fwCanonicalPoints`, `fwSanitizeLines`, etc. available. Then:

1. Evaluate the example's geometry-generator cell — it writes the `.wl` file.
2. In the *File* tab, **Load .wl** that file.
3. Assign sections in the *Properties* tab and apply the loads listed under each example's "Applying loads" steps.
4. *Results* tab → select a load case → **Run FEA**.

> If `Framica[]` comes up blue/unevaluated after a restart, load the paclet first (`Needs[...]`) in its own cell; loading it also clears any stray `Global`Framica` that the front end may have created.

Chapter 14

Worked Example 1 — Portal Frame



14.1 Description

A portal frame is a rigid plane frame of two vertical columns connected by a beam or pitched rafters, with moment-resisting joints at the eaves. This example covers three variants:

- **Part A** — Rectangular portal, fixed bases. Full hand-calculation verification.
- **Part B** — Pitched portal, two inclined rafters.
- **Part C** — 3D multi-bay industrial shed.

Since the geometry lies in the XZ plane, set the viewport to **ElY** for a standard structural elevation view.

14.2 Part A — Rectangular Portal Frame

14.2.1 Geometry

14.2.2 Geometry generator

Listing 14.1: Rectangular portal frame generator

Table 14.1: Rectangular portal frame geometry ($L = 6\,\text{m}$, $h = 4\,\text{m}$)

Node	X	Y	Z	Description
1	0	0	0	Left base (fixed support)
2	6	0	0	Right base (fixed support)
3	0	0	4	Left eaves
4	6	0	4	Right eaves

Table 14.2: Portal frame members

Member	From → To	Description
1	1 → 3	Left column
2	2 → 4	Right column
3	3 → 4	Rafter / beam

```
Module[{lines, pts, nodeProps, loadCases},

  lines = {
    {{0., 0., 0.}, {0., 0., 4.}},    (* left column  *)
    {{6., 0., 0.}, {6., 0., 4.}},    (* right column *)
    {{0., 0., 4.}, {6., 0., 4.}}     (* beam         *)
  };

  pts = fwCanonicalPoints[fwSanitizeLines[lines]];

  nodeProps = Association @ Table[
    If[pts[[i,3]] < 0.01,
      i -> <|"Release" -> "Fixed"|>, Nothing],
    {i, Length[pts]}];

  loadCases = <|
    "G_dead" -> <|"Type"->"Primary","Loads"->{}|>,
    "Q_imp"  -> <|"Type"->"Primary","Loads"->{}|>,
    "W_wind" -> <|"Type"->"Primary","Loads"->{}|>,
    "ULS"    -> <|"Type"->"Combination",
      "Components"-><|"G_dead"->1.35,"Q_imp"->1.5|>|>
  |>;

  fwSaveProject[
    FileNameJoin[{$HomeDirectory,"Desktop","portal_frame.wl"}],
    lines, pts, <||>, nodeProps, <||>, loadCases,
    <|"EditMode"->"Members","ShowPtLabels"->True,
      "ShowLnLabels"->True,"PlotSize"->600,
      "CurrentLoadCase"->"G_dead"|>]
]
```

14.2.3 Applying loads

G_dead Select beam (member 3). UDL, Global: $w_z = -10\,\text{kN/m}$, $w_x = w_y = 0$.

Q_imp Select beam. UDL, Global: $w_z = -5\,\text{kN/m}$.

W_wind Nodes mode; select left eaves node (node 3). Point Force, Global: $F_x = +20\,\text{kN}$.

ULS Combination: $1.35\,G_{\text{dead}} + 1.5\,Q_{\text{imp}}$.

14.2.4 Verification formulas (fixed base, same EI, UDL w on beam)

For a fixed-base symmetric rectangular portal frame with identical EI throughout, span L, column height h, and UDL w on the beam:

$$M_{\text{eaves}} = \frac{wL^3}{6(2L+h)} \tag{14.1}$$

$$M_{\text{base}} = \frac{M_{\text{eaves}}}{2} = \frac{wL^3}{12(2L+h)} \tag{14.2}$$

$$M_{\text{mid}} = \frac{wL^2(2L+3h)}{24(2L+h)} \tag{14.3}$$

$$H = \frac{wL^3}{4h(2L+h)} \tag{14.4}$$

$$\delta_{\text{mid}} = \frac{1}{EI}\left(\frac{5wL^4}{384} - \frac{M_{\text{eaves}}\,L^2}{8}\right) \tag{14.5}$$

Table 14.3: Verification values for $w = 10\,\text{kN/m}$, $L = 6\,\text{m}$, $h = 4\,\text{m}$, IPE 240 ($EI = 8173\,\text{kNm}^2$)

Quantity	Formula	Value	FEA target
M_{eaves} (hogging)	Eq. (14.1)	22.5 kNm	22.5 kNm
M_{base} (hogging)	Eq. (14.2)	11.25 kNm	11.25 kNm
M_{mid} (sagging)	Eq. (14.3)	22.5 kNm	22.5 kNm
H (horizontal thrust)	Eq. (14.4)	8.44 kN	8.44 kN
V (vertical reaction)	$wL/2$	30.0 kN	30.0 kN
δ_{mid}	Eq. (14.5)	8.3 mm	≈8.3 mm

Coincidence: For this geometry ($L/h = 3/2$), the eaves and midspan moments happen to be equal because $3h = 2L$. For any other L/h ratio use the general formulas above.

14.3 Part B — Pitched Portal Frame

14.3.1 Geometry changes

Listing 14.2: Pitched portal geometry

```
lines = {
  {{0., 0., 0.}, {0., 0., 4.}},     (* left column  *)
  {{8., 0., 0.}, {8., 0., 4.}},     (* right column *)
  {{0., 0., 4.}, {4., 0., 5.5}},    (* left rafter  *)
  {{4., 0., 5.5}, {8., 0., 4.}}     (* right rafter *)
};
```

Table 14.4: Pitched portal frame nodes ($L = 8\,\text{m}$, $h = 4\,\text{m}$, ridge $= 5.5\,\text{m}$)

Node	X	Y	Z	Description
1	0	0	0.0	Left base
2	8	0	0.0	Right base
3	0	0	4.0	Left eaves
4	8	0	4.0	Right eaves
5	4	0	5.5	Ridge (pitch $\approx 20.6°$)

14.3.2 Key differences from the rectangular variant

Table 14.5: Rectangular vs. pitched portal comparison

Aspect	Rectangular	Pitched
Rafter axial force	Near zero (pure bending)	Non-zero compression — partly arch-like
Horizontal base thrust	Lower	Higher for same span
Snow load direction	Vertical = perpendicular to beam	Distinguish vertical vs. slope-normal
Ridge connection	Not applicable	Moment-resisting; pinned ridge creates a mechanism

14.4 Part C — 3D Two-Bay Industrial Shed

Listing 14.3: 3D portal shed generator

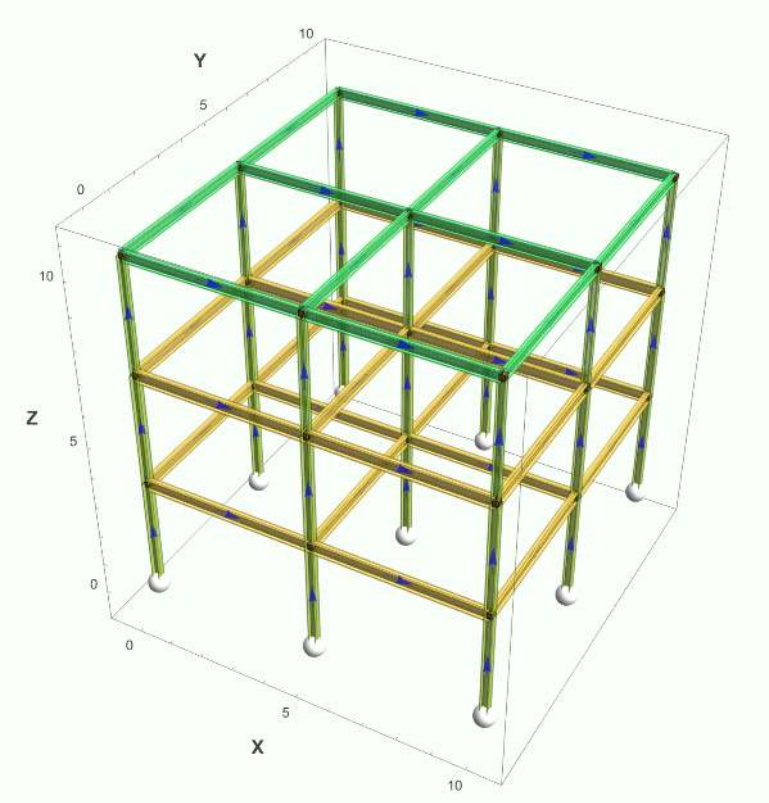
```
Module[{
  L = 8.0, h = 4.0, hr = 5.5,
  bay = 6.0, nBays = 2,
  lines, pts, nodeProps, loadCases
},
  lines = Flatten[Table[
    { {{0.,k*bay,0.},{0.,k*bay,h}}, {{L,k*bay,0.},{L,k*bay,h}},
      {{0.,k*bay,h},{L/2,k*bay,hr}}, {{L/2,k*bay,hr},{L,k*bay,h}} },
    {k, 0, nBays}], 2];

  Do[
    AppendTo[lines, {{0.,k*bay,h},{0.,(k+1)*bay,h}}];
    AppendTo[lines, {{L, k*bay,h},{L, (k+1)*bay,h}}];
    AppendTo[lines, {{L/2,k*bay,hr},{L/2,(k+1)*bay,hr}}],
    {k, 0, nBays-1}];

  AppendTo[lines, {{0.,0.,0.},{0.,bay,h}}];
  AppendTo[lines, {{L, 0.,0.},{L, bay,h}}];

  pts = fwCanonicalPoints[fwSanitizeLines[lines]];
  nodeProps = Association @ Table[
    If[pts[[i,3]] < 0.01, i -> <|"Release"->"Fixed"|>, Nothing],
    {i, Length[pts]}];

  fwSaveProject[
```

```
    FileNameJoin[{$HomeDirectory,"Desktop","portal_shed_3D.wl"}],
    lines, pts, <||>, nodeProps, <||>,
    <|"G_dead"-><|"Type"->"Primary","Loads"->{}|>,
      "Q_roof" -><|"Type"->"Primary","Loads"->{}|>,
      "W_windY"-><|"Type"->"Primary","Loads"->{}|>|>,
    <|"PlotSize"->650,"CurrentLoadCase"->"G_dead"|>]
]
```

Bracing bay: Without the wall bracing members, the frame has no lateral stiffness in Y and FEA will fail with an instability error. This demonstrates the essential role of a bracing bay.

Chapter 15

Worked Example 2 — Multi-Storey Moment Frame



15.1 Description

A multi-storey moment frame resists gravity and lateral loads through bending and shear in columns and beams, with moment-resisting connections at every joint. This example models a 2×2 bay, 3-storey steel frame.

15.2 Geometry Parameters

15.3 Geometry Generator

Listing 15.1: Multi-storey moment frame generator

Table 15.1: Moment frame parameters

Parameter	Value	
Bays in X (n_x)	2	
Bays in Y (n_y)	2	
Bay width $L_x = L_y$	5.0 m	
Storey height h_s	3.5 m	
Storeys n_s	3	Total height 10.5 m
Total nodes	36	$(n_x+1)(n_y+1)(n_s+1)$
Total members	63	27 columns + 18 X-beams + 18 Y-beams

```
Module[{
  nx = 2, Lx = 5.0, ny = 2, Ly = 5.0, ns = 3, hs = 3.5,
  pt, lines, pts, nodeProps, loadCases
},
  pt[i_, j_, k_] := N @ {i*Lx, j*Ly, k*hs};

  lines = Flatten[{
    Table[{pt[i,j,k], pt[i,j,k+1]},
      {k,0,ns-1},{i,0,nx},{j,0,ny}],
    Table[{pt[i,j,k], pt[i+1,j,k]},
      {k,1,ns},{j,0,ny},{i,0,nx-1}],
    Table[{pt[i,j,k], pt[i,j+1,k]},
      {k,1,ns},{i,0,nx},{j,0,ny-1}]
  }, 3];

  pts = fwCanonicalPoints[fwSanitizeLines[lines]];
  nodeProps = Association @ Table[
    If[pts[[i,3]] < 0.01, i -> <|"Release"->"Fixed"|>, Nothing],
    {i, Length[pts]}];

  loadCases = <|
    "G_dead"  -> <|"Type"->"Primary","Loads"->{
      <|"Type"->"Gravity","gx"->0,"gy"->0,"gz"->-9.81|>}|>,
    "Q_imp"   -> <|"Type"->"Primary","Loads"->{}|>,
    "Wx_wind" -> <|"Type"->"Primary","Loads"->{}|>,
    "ULS_GQ"  -> <|"Type"->"Combination",
      "Components"-><|"G_dead"->1.35,"Q_imp"->1.5|>|>,
    "ULS_GQW" -> <|"Type"->"Combination",
      "Components"-><|"G_dead"->1.35,"Q_imp"->1.05,"Wx_wind"->1.5|>|>
  |>;

  fwSaveProject[
    FileNameJoin[{$HomeDirectory,"Desktop","moment_frame_3D.wl"}],
    lines, pts, <||>, nodeProps, <||>, loadCases,
    <|"PlotSize"->700,"CurrentLoadCase"->"G_dead"|>]
]
```

15.4 Section Assignment

Table 15.2: Suggested sections for the 3D moment frame

Group	Level (z, m)	Section	Notes
`col_S1`	$0 \to 3.5$	HEA 240	Highest demand
`col_S2`	$3.5 \to 7.0$	HEA 240	Match S1 for regularity
`col_S3`	$7.0 \to 10.5$	HEA 200	Reduced load at top
`beams_L1`	Level 1 ($z = 3.5$)	IPE 360	
`beams_L2`	Level 2 ($z = 7.0$)	IPE 360	
`beams_roof`	Roof ($z = 10.5$)	IPE 300	Lighter imposed load

15.5 Wind Verification — Portal Method

The *Portal Method* gives approximate column shears and beam end moments for multi-storey frames. Each of the three Y-frame lines carries 1/3 of the total horizontal force: $45\,\text{kN}/3 = 15\,\text{kN}$ per floor per frame.

Table 15.3: Portal Method — storey shears ($F = 15\,\text{kN/floor}$, $n = 2$ bays)

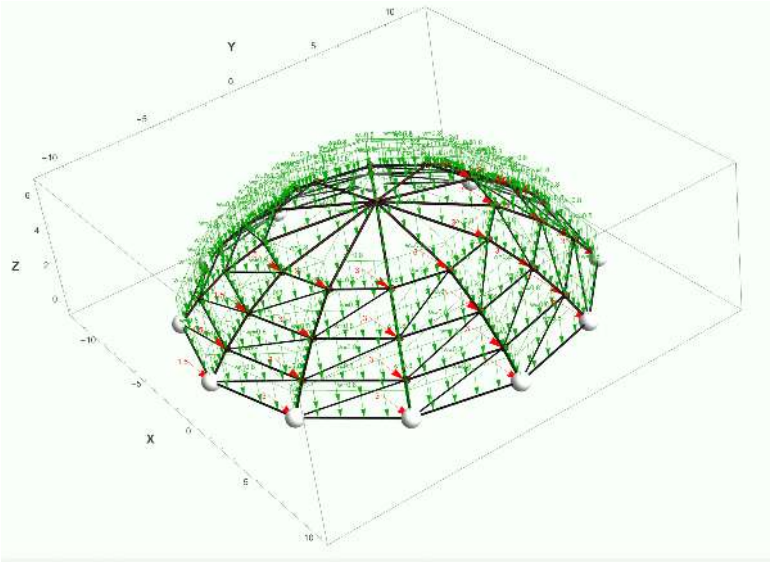
Storey	V_k (kN)	Ext. shear (kN)	Int. shear (kN)
3 (top)	15	3.75	7.50
2	30	7.50	15.0
1	45	11.25	22.5

Table 15.4: Portal Method — column and beam end moments ($h_s/2 = 1.75\,\text{m}$)

Storey	M_{ext} (kNm)	M_{int} (kNm)	Level	$M_{\text{beam end}}$ (kNm)
3	$3.75 \times 1.75 = 6.6$	$7.5 \times 1.75 = 13.1$	Roof	6.6
2	$7.5 \times 1.75 = 13.1$	$15.0 \times 1.75 = 26.3$	Level 2	19.7
1	$11.25 \times 1.75 = 19.7$	$22.5 \times 1.75 = 39.4$	Level 1	32.8

Chapter 16

Worked Example 3 — Schwedler Dome



16.1 Description

A Schwedler dome is a single-layer lattice dome with meridional ribs, circumferential rings, and diagonal bracing.

16.2 Geometry Parameters

Table 16.1: Schwedler dome parameters

Parameter	Value	Notes
Base radius R	10.0 m	
Crown height H	5.0 m	Rise/span = 0.25
Sphere radius r_s	12.5 m	$r_s = (R^2 + H^2)/(2H)$
Meridians n	12	30° spacing
Intermediate rings	3	Plus base and apex
Total nodes	49	
Total members	132	

The ring radii follow from the sphere equation:

$$r(z) = \sqrt{r_s^2 - (z + r_s - H)^2}$$

16.3 Geometry Generator

Listing 16.1: Schwedler dome geometry generator

```
Module[{
  R=10.0, H=5.0, n=12, nR=3,
  rs, zLev, rLev, co, apx, lines, pts, nodeProps, loadCases
},
  rs   = (R^2 + H^2)/(2 H);
  zLev = N @ Table[k H/(nR+1), {k, 0, nR+1}];
  rLev = N @ Sqrt[rs^2 - (zLev + rs - H)^2];

  co[lev_, k_] := {rLev[[lev+1]] Cos[2 Pi k/n],
                   rLev[[lev+1]] Sin[2 Pi k/n], zLev[[lev+1]]};
  apx = {0., 0., H};
  lines = {};

  Do[Do[AppendTo[lines, N@{co[lev,k], co[lev+1,k]}],
        {k,0,n-1}], {lev,0,nR-1}];
  Do[AppendTo[lines, N@{co[nR,k], apx}], {k,0,n-1}];
  Do[Do[AppendTo[lines, N@{co[lev,k], co[lev,Mod[k+1,n]]}],
        {k,0,n-1}], {lev,0,nR}];
  Do[Do[AppendTo[lines, N@{co[lev,k], co[lev+1,Mod[k+1,n]]}],
        {k,0,n-1}], {lev,0,nR-1}];

  pts = fwCanonicalPoints[fwSanitizeLines[lines]];
  nodeProps = Association @ Table[
    If[pts[[i,3]] < 0.01, i -> <|"Release"->"Pinned"|>, Nothing],
    {i, Length[pts]}];

  loadCases = <|
    "G_self" -> <|"Type"->"Primary","Loads"->{
      <|"Type"->"Gravity","gx"->0,"gy"->0,"gz"->-9.81|>}|>,
    "Q_snow" -> <|"Type"->"Primary","Loads"->{}|>,
    "W_windX"-> <|"Type"->"Primary","Loads"->{}|>,
    "ULS_GS" -> <|"Type"->"Combination",
      "Components"-><|"G_self"->1.35,"Q_snow"->1.5|>|>
  |>;

  fwSaveProject[
    FileNameJoin[{$HomeDirectory,"Desktop","schwedler_dome.wl"}],
    lines, pts, <||>, nodeProps, <||>, loadCases,
    <|"PlotSize"->650, "CurrentLoadCase"->"G_self"|>]
]
```

16.4 Verification

For a uniform vertical load q on a spherical dome of sphere radius r_s, the meridional thrust per unit length at the base ring is:

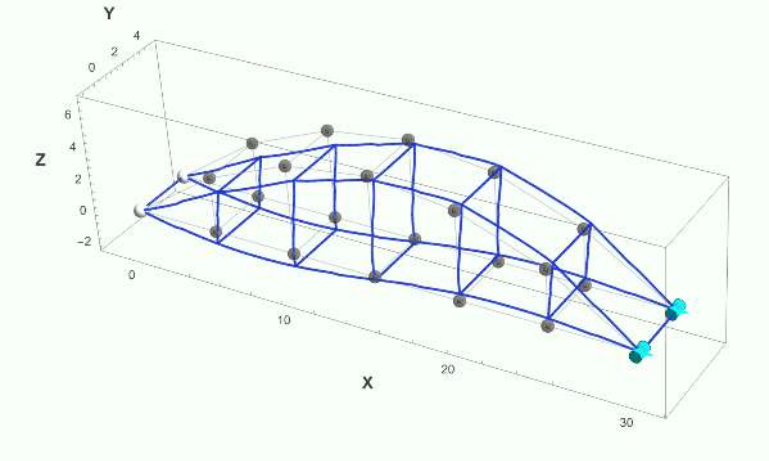
$$N_{\text{merid}} = \frac{-q\, r_s}{1 + \cos\phi_{\text{base}}}$$

For $q \approx 0.6\,\text{kN/m}^2$ and $\phi_{\text{base}} \approx 53°$:

$$N_{\text{merid}} \approx \frac{-0.6 \times 12.5}{1.6} \approx -4.7\,\text{kN/m} \quad \Longrightarrow \quad \approx -24\,\text{kN per base member}$$

Chapter 17

Worked Example 4 — Arch Bridge with Suspended Deck



17.1 Description

A through tied arch bridge combines arch ribs (compression), hangers (tension), and deck beams (tie in tension). A parabolic arch under UDL carries load in pure axial compression with zero bending — providing an exact analytical verification target.

17.2 Geometry Parameters

Table 17.1: Arch bridge parameters

Parameter	Symbol	Value
Span	L	30.0 m
Arch rise	f	6.0 m ($f/L = 1/5$)
Bridge width	W	4.0 m
Panels	n_P	6 (hanger spacing 5 m)
Total nodes		24
Total members		46

The arch profile:

$$z_{\text{arch}}(x) = \frac{4f}{L^2}\, x\,(L - x)$$

17.3 Geometry Generator

Listing 17.1: Through arch bridge generator

```
Module[{
  L=30.0, f=6.0, W=4.0, nP=6,
  dx, xPos, archZ, archPt, deckPt, lines, pts, nodeProps, loadCases
},
  dx    = L/nP;
  xPos  = N @ Table[k*dx, {k, 0, nP}];
  archZ = N @ (4 f/L^2 * # * (L - #))& /@ xPos;

  archPt[k_, y_] := {xPos[[k+1]], N@y, archZ[[k+1]]};
  deckPt[k_, y_] := {xPos[[k+1]], N@y, 0.};
  lines = {};

  Do[Do[AppendTo[lines, {archPt[k,y], archPt[k+1,y]}],
        {k,0,nP-1}], {y,{0.,W}}];
  Do[Do[AppendTo[lines, {deckPt[k,y], deckPt[k+1,y]}],
        {k,0,nP-1}], {y,{0.,W}}];
  Do[Do[AppendTo[lines, {archPt[k,y], deckPt[k,y]}],
        {k,1,nP-1}], {y,{0.,W}}];
  Do[AppendTo[lines, {deckPt[k,0.], deckPt[k,W]}], {k,0,nP}];
  Do[AppendTo[lines, {archPt[k,0.], archPt[k,W]}], {k,1,nP-1}];

  pts = fwCanonicalPoints[fwSanitizeLines[lines]];
  nodeProps = Association @ Table[
    If[pts[[i,3]] < 0.01 &&
       (pts[[i,1]] < 0.01 || pts[[i,1]] > L-0.01),
      i -> <|"Release"->"Pinned"|>, Nothing],
    {i, Length[pts]}];

  loadCases = <|
    "G_self"   -> <|"Type"->"Primary","Loads"->{
      <|"Type"->"Gravity","gx"->0,"gy"->0,"gz"->-9.81|>}|>,
    "G_dead"   -> <|"Type"->"Primary","Loads"->{}|>,
    "Q_full"   -> <|"Type"->"Primary","Loads"->{}|>,
    "Q_patch"  -> <|"Type"->"Primary","Loads"->{}|>,
    "ULS_GQ"   -> <|"Type"->"Combination",
      "Components"-><|"G_self"->1.35,"G_dead"->1.35,"Q_full"->1.5|>|>,
    "ULS_patch"-> <|"Type"->"Combination",
      "Components"-><|"G_self"->1.35,"G_dead"->1.35,"Q_patch"->1.5|>|>
  |>;

  fwSaveProject[
    FileNameJoin[{$HomeDirectory,"Desktop","arch_bridge.wl"}],
    lines, pts, <||>, nodeProps, <||>, loadCases,
    <|"ShowPtLabels"->True,"PlotSize"->600,
      "CurrentLoadCase"->"G_dead"|>]
]
```

Important: Set the **hangers** group Force Type to **Tension Only** *before* running FEA. Under patch loading some hangers may go slack (zero force) — this is physically correct behaviour.

17.4 Verification — Parabolic Arch Under UDL

For one arch rib, $w = 10\,\text{kN/m}$, $L = 30\,\text{m}$, $f = 6\,\text{m}$:

$$H = \frac{wL^2}{8f} = \frac{10 \times 900}{48} = 187.5\,\text{kN} \qquad V = \frac{wL}{2} = 150\,\text{kN}$$

$$N(x) = -\sqrt{H^2 + V(x)^2}, \qquad V(x) = w\left(\tfrac{L}{2} - x\right)$$

Table 17.2: Arch axial force at panel points (one rib, $w = 10\,\text{kN/m}$)

x (m)	$V(x)$ (kN)	$N(x)$ (kN)	Comment
0	150	−240.1	Max compression (springing)
5	100	−213.5	
10	50	−194.1	
15	0	−187.5	Crown; equals H
20	−50	−194.1	
25	−100	−213.5	
30	−150	−240.1	

Deck tie tension: $N_{\text{deck}} = +H = +187.5\,\text{kN}$ (uniform). Hanger forces: $F_h = w \cdot dx = 10 \times 5 = 50\,\text{kN}$ each (equal under UDL). Support $F_x \approx 0\,\text{kN}$ — the key tied-arch check.