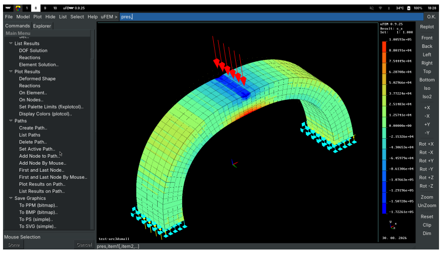
uFEM Software Suite

User Guide for `gfem`, `gfbrowse`, `goutview`, and `fem`

Jiri Brozovsky
AI Engineering Lead: Etsuko

August 30, 2026



Abstract

This comprehensive user guide provides a manual for the **uFEM** (Micro Finite Element Method) suite, covering the graphical pre/post-processor `gfem`, database browser `gfbrowse`, output text viewer `goutview`, terminal variant `tfem`, terminal viewer `toutview`, and the standalone matrix solver engine `fem`. Detailed explanations of geometric pre-processing, meshing, material models, boundary conditions, solution strategies, result visualization (`plns` vs. `ples`), and path cutting are presented. Five fully verified real-world engineering benchmark macro examples with structural schematics and macro listings are included.

Contents

1 Introduction

1.1 Basic Program Structure and Ecosystem Architecture

The **uFEM** finite element analysis system is designed as s lightweight software written in the ANSI C89 with almost zero 3rd-party library dependencies [?, ?]. The Graphical User Interface (GUI) and the 3D canvas (used for visual presentation of data) uses primarily the Gtk+-2.0.x library and the OpenGL 1.1 interface. That is, the system is expected to run on older UNIX and Linux systems from mid- to late- 2000s. For special cases there is a (fuctionally limited) Gtk+-1.2 GUI (uses the GtkGLArea library) or a very simpplified GLUT-based interface. In any case, all of these interfaces can utilise the same command language and data formats.

A portable web-based interface is in development. This interface is portable on cost of effectivity.

The ecosystem consists of several specialized executables operating cooperatively:

- `gfem`: The primary graphical pre-processor, interactive modeler, and post-processor built using GTK+ 2.0 and OpenGL 1.1 graphics.
- `gfbrowse`: A graphical file and project database browser for quickly managing, inspecting, and launching uFEM job files.
- `goutview`: A GTK+ multi-tab text output viewer designed for real-time monitoring of solver output files (`.res`, `.out`, `.log`) with automatic tab management.
- `fem`: The standalone, high-performance C finite element solver executable. Must always be launched with the `-p` parameter (`fem -p jobname`) [?].
- `tfem`: The terminal VT100 interactive Pre/Post-processor for headless servers and batch console scripting. No graphics nor image outputs are available here.
- `toutview`: The terminal VT100 text file output viewer.

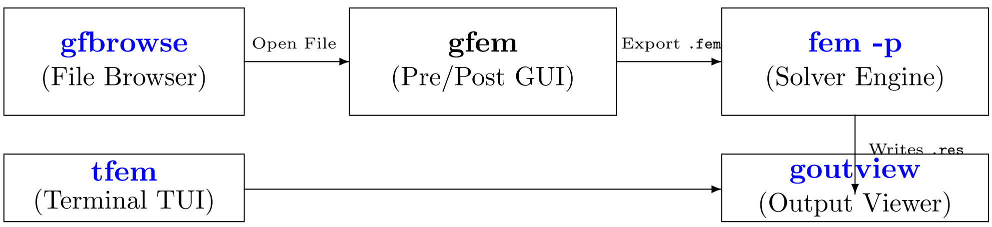


Figure 1: uFEM Software Suite Component Architecture and Data Flow.

1.2 gfem Analysis Capabilities, Simplifications, and Limitations

The `gfem` solver engine supports a wide range of structural, thermal, and multiphysics analysis categories [?, ?]:

1. **Linear Static Analysis**: 1D beams/trusses, 2D plane stress/strain/axisymmetric, 3D solids, plates, and shells.
2. **Non-Linear Material & Geometric Analysis**: Von Mises plasticity, Drucker-Prager, Chen-Chen concrete, Hill anisotropic yield, strain hardening (Hollomon/Ludwik), and arc-length path following.

3. **Structural Dynamics & Modal Analysis**: Eigenfrequency analysis and direct time integration (Newmark/Wilson-θ).
4. **Structural Stability**: Eigenvalue buckling and non-linear post-buckling.
5. **Thermal Analysis**: 2D and 3D steady-state heat conduction and thermal stress coupling.
6. **Contact Mechanics**: 2D and 3D node-to-segment and segment-to-segment contact pairs.
7. Basic **Fatigue Damage Prediction**: Low-cycle and high-cycle fatigue life calculation.

Limitations: All code is written strictly in ANSI C89. Dynamic memory allocation is managed through `fdb` memory structures. OpenGL functions are strictly bounded to OpenGL 1.1 to ensure compatibility on legacy hardware without requiring modern shader pipelines. These features go at cost of computational and visualizational speed and limit scalability.

1.3 Model Import and Export Formats

Please note that external formats are usually limited and not all available models can be imported.

- **uFEM Macro (`.mac`)**: Standard procedural command text file.
- **ANSYS Finite Element Database (`.cdb`)**: Partial import and export compatibility for nodes, elements, and materials.
- **Stereolithography CAD (`.stl`)**: Import and export of 3D triangular surface meshes.
- **Drawing Exchange Format (`.dxf`)**: Import of 2D CAD boundary curves.

1.4 Supported Target Operating Systems & Feature Availability

Table 1: Platform Compatibility Matrix across uFEM Software Suite Components.

Target Operating System / Platform	`gfem`	`gfbrowse`	`goutview`	`fem`	`tfem`	**gfem_web**
Linux (X11 / GTK+ 2.0)	Supported	Supported	Supported	Supported	Supported	Supported
SGI IRIX 6.5	Gtk+-1.0 only	N/A	N/A	Supported	Supported	N/A
SGI IRIX 5.3 / 6.5	Supported	Supported	Supported	Supported	Supported	Supported
Windows (Win32 Console)	N/A (Web UI)	N/A	N/A	Supported	Supported	Supported (`.exe`)
WebAssembly (Browser)	Web GUI	Web Explorer	In-Browser Log	WASM Engine	N/A	Native WASM
Android (Capacitor App)	Touch Web GUI	Touch Explorer	Log Drawer	WASM Engine	N/A	Built-in App

Limitations: The `gfem_web` can run on a legacy systems like the IRIX flawlesly but its web-based interface must be accessed from modern WWW browser with contemporary standards support. Such browsers **do not exist** for most of legacy systems like IRIX, AIX, PLAN9 and others. Moreover, the `gfem_web` has still quite limited GUI capabilities and many commands must be entered via the command line.

2 gfbrowse - Graphical File and Database Browser

2.1 Program Description, Controls, and Safeguards

gfbrowse is a lightweight GTK+ application designed to manage uFEM macro files (`.mac`), finite element input decks (`.fem`), and result files (`.res`). The **gfbrowse** directly views `.db` files, the `.fem` files are first converted to viewable form and the `.mac` (macro) files are executed on request (to prevent long waiting times for complex macros).

- **Function**: Directory navigation, file previews, one-click launcher for `gfem` and goutview.
- **Controls**: Folder tree on the left panel, file details in the central view, quick action toolbar at the top.
- **Safeguards**: Prevents overwriting active files, validates `.mac` command syntax before launching execution.

2.2 Model viewer feature

The program allows to browse directories. If a compatible format is selected then the program tries to obtain data and show basic geometry and textual summary of the model. In case of `.mac` files is only interpretes a part of script. For more complex and parametrc scripts a `double click` may be used to initiate interpretation of the script up to the first `mesh` command to obtain necessary data. The complete `.mac` script is shown at the bottom of the textual summary.

2.3 Interaction with gfem

The procedure slightly depends on the file format:

- Selecting a `.mac` macro file in gfbrowse and clicking **Run in gfem** invokes `gfem -b filename.mac`, initiating graphical visualization and execution of the whole script. This may be time consuming, espeially if there are non-linear solution commands or large parametric models.
- If a `.fem` file is opened with **Run in gfem** then an auxiliary script is called to convert it to the native `.db` database format. Then the `.db` is opened in the `gfem`. The `.db` is opened directly. In case of these two formats the `gfem` is opened in the preprocessor and waits here (these formats do not store instructions to execute solvers nor to work with results).
- If a `.res` (results) file is opened in this way then the program tries to locate the compatible `.db` or the `.fem` file to reconstruct the geometry and to allows user to see the results.

3 goutview - Output File Viewer

3.1 Program Description, Controls, and Safeguards

goutview is a GTK+ 2.0 output text viewer featuring multi-tab document rendering, realtime syntax highlighting for solver keywords, and UNIX domain socket IPC for single-instance operations.

This program can be started before the `gfem` and is persistent. The `gfem` fills it with output data when one of *print* commands (`pr...`, like `prns`) is executed. Alternative program for this purpose can be configured in the `.tgfemrc` configuration file or in `gfem`/`tfem` command line with use of the `outview,PROGRAM` command.

- **Tab Limits**: Supports up to **128 open tabs**.
- **Auto-FIFO Rotation**: When the maximum open tabs limit (128 tabs) is reached, `goutview` automatically closes the oldest tab (`close_tab(0)`), allowing seamless continuous file opening.
- **Keyboard Shortcuts**:
  - `Ctrl+O`: Open File dialog.
  - `Ctrl+W`: Close Active Tab.
  - `Ctrl+F`: Focus Search Bar.
  - `F3`: Find Next search match.
  - `Ctrl+R`: Reload Active File.
  - `Ctrl+Q`: Quit Viewer.

4 gfem - Pre/Post-processor Pre/Post Processing Suite

4.1 Program Setup and Configuration File (`.tgfemrc`)

When starting `gfem`, configuration parameters are read sequentially from:

1. `$HOME/.tgfemrc`
2. `./tgfemrc`
3. `/usr/share/ufem/tgfemrc`

4.1.1 Commented Example Configuration File

```
# uFEM Configuration File (.tgfemrc)
# Color scheme definitions (R, G, B floats 0.0 - 1.0)
plotcol, background, 1.0, 1.0, 1.0
plotcol, mesh, 0.2, 0.2, 0.2
plotcol, nodes, 0.0, 0.4, 0.8
plotcol, loads, 0.9, 0.1, 0.1

# Path to standalone solver binary
setsolver, /usr/local/bin/fem

# Temporary data directory for solver export files
datadir, /tmp

# Number of parallel threads for pthreads solver (GUI, solver)
nproc, 4, 4

# Output viewer configuration
outview, goutview
outauto, 1
```

Listing 1: Commented `.tgfemrc` configuration file example.

4.2 Standard Modeling and Analysis Workflow

The workflow in `gfem` follows a 7-step sequence:

1. **Initialization**: Clear data (`cleardata`) and set job name (`jobname,myjob`).
2. **Properties**: Define material properties (`mat`, `mp`) and element formulations (`et`, `rs`, `cs`). You may use the material selection helper from the `Model` menu in the main menubar. In case of beam elements you can use the cross-section helper from the same menu.
3. **Geometry & Meshing**: Create keypoints (`k`), lines (`ge`), areas (`a`), set divisions (`gediv`), and mesh (`mesh`). There are functions to create some basic geometric entities (rectangles, cubes and other). In this case the needed keypoints are generated automaticaly.
4. **Boundary Conditions & Loads**: Apply constraints (`d`), point loads (`f`), element loads (`el`), and gravity (`accel`).
5. **Solution**: Run static or dynamic solver (`solve,linear`).
6. **Post-Processing**: Plot displacements (`pldef`), nodal stresses (`plns`), or element forces (`ples`).
7. **Image & Report Export**: Save graphics (`gl2bmp,out.bmp`, `gl2ps,out.ps`) and generate reports (`report,summary`).

4.3 Basic Command Reference Table

Table 2: Essential uFEM Commands Summary.

Command	Parameters	Description
`cleardata`	None	Resets active finite element database in memory
`jobname`	`name`	Sets current analysis job name string
`help`	`[command]`	Displays interactive help and syntax for uFEM commands
`mat`	`id, type`	Activates material set ID
`mp`	`prop, mat, val`	Sets material property (`ex`, `prxy`, `dens`, `fyc`)
`et`	`id, type`	Sets element type formulation ID
`rs`	`id, type`	Sets real constants set ID
`cs`	`section, set, type`	Sets standard cross-section profile (e.g. `IPE 300`)
`k`	`id, x, y, z`	Defines geometric keypoint
`n`	`id, x, y, z`	Defines finite element node
`e`	`id, n1, n2, ...`	Defines finite element by node connection list
`d`	`node, dof, val`	Applies displacement constraint (`ux`, `uy`, `uz`, `rotx`)
`f`	`node, dof, val`	Applies nodal force/moment load
`el`	`elem, type, val`	Applies distributed element line load (`qy`, `qz`)
`solve`	`type, iters, limit`	Exports `.fem` file and invokes solver `fem -p`
`solve,coupled`	`[tempfile]`	Runs 2-pass coupled thermal → structural solver (`tsolve`)
`plns`	`item`	Plots smoothed nodal stress field (`s_vmis`, `s_x`)
`ples`	`item`	Plots element diagram/result (`m_z`, `f_y`, `n_x`)
`clip`	`subcmd, [val|A,B,C,D]`	Activates 3D clipping cut plane (`x`, `y`, `z`, `coefs`, `off`)
`path`	`[num, name]`	Initializes an active path container
`ptwo`	`n1, n2, div`	Adds cut line segment between nodes N_1, N_2 with `div` divisions
`pn`	`node`	Adds a single node to active path
`plpath`	`item, zoom`	Plots 2D graph of results along the active 1D path
`prpath`	`item1, item2`	Lists result values at all path nodes to output
`gl2bmp`	`filename`	Captures active OpenGL viewport to BMP bitmap

4.3.1 Interactive Help System (`help`)

uFEM includes an interactive built-in help system accessible at any time from both `gfem` and `tfem`. The `help` command supports several query modes:

- `help` (without parameters): Displays active system status, including uFEM version, total element formulations available, total material models available, current workspace mode (Preprocessing or Postprocessing), active job name, and data directory path.
- `help, list` (or `help, commands`): Prints a complete alphabetical directory of all interactive commands supported by the command interpreter.
- `help, <command>` (e.g. `help, solve`, `help, geextrude`): Prints exact syntax, required or optional parameters, and usage hints for the specified command.
- `help, e, <type_id>` (e.g. `help, e, 26`): Displays complete element type formulation specifications, including required node counts, degrees of freedom (DOFs), static real constants, repeating parameters, and result quantities. Specify `help, e, all` to inspect all element types.
- `help, m, <type_id>` (e.g. `help, m, 1`): Displays material model definitions, listing required properties (e.g. Young's modulus `ex`, Poisson's ratio `prxy`, density `dens`, yield strength `fyc`) and repeating values. Specify `help, m, all` to inspect all material models.

4.4 Result Post-Processing: `plns` vs. `ples`, Path Cuts, and 3D Model Clipping (`clip`)

4.4.1 Contour Plots

There are two main types of contour (color map) plots:

- `plns` **(Plot Nodal Solution)**: Computes stress tensors at element integration points, extrapolates to nodes, and averages values across shared nodes to form continuous contour maps (σ_{vm}, σ_x, σ_y).
- `ples` **(Plot Element Solution)**: Displays un-averaged, discrete elemental values. Crucial for 1D beam elements to plot exact bending moment (M_z), shear force (Q_y), and axial force (N_x) diagrams without smoothing discontinuities at joints.

4.4.2 3D Model Section Cuts with OpenGL Clipping Planes (`clip`)

For complex 3D solid or shell structures (such as cooling towers, solid blocks, or pressure vessels), interior stresses cannot be visualized from external surface contours alone. `gfem` provides a dedicated OpenGL hardware-accelerated clipping plane feature controlled via the `clip` command:

- `clip, x, val`: Cuts the 3D model with a vertical clipping plane at coordinate $X = \text{val}$ (plane equation $1 \cdot X + 0 \cdot Y + 0 \cdot Z - \text{val} = 0$).
- `clip, y, val`: Cuts the 3D model with a horizontal plane at $Y = \text{val}$.
- `clip, z, val`: Cuts the 3D model with a depth plane at $Z = \text{val}$.
- `clip, coefs, A, B, C, D`: Defines an arbitrary oblique 3D section plane given by coefficients $A \cdot X + B \cdot Y + C \cdot Z + D = 0$.
- `clip, off`: Disables the clipping plane and restores full 3D model visibility.
- `clip, on`: Re-enables the active clipping plane.

```
! Enable section cut at X = 0 to reveal interior solid stress distribution
clip, x, 0.0
plns, s\_vmis
gl2bmp, tower\_interior\_cut.bmp

! Turn off clipping plane
clip, off
```

4.4.3 1D Section Path Cuts (`path`, `ptwo`, `pn`, `plpath`, `prpath`)

To obtain precise quantitative stress or displacement profiles along a linear cut line across a 2D or 3D domain:

1. Initialize a new active path container: `path, 1, CUT1`.
2. Add path nodes along a straight section cut line between node N_1 and node N_2 with 10 subdivisions: `ptwo, n1, n2, 10` (or add individual nodes sequentially using `pn, node_id`).
3. Plot the result variable profile along the path distance: `plpath, s_vmis, 1.0`.
4. Alternatively, print a tabular report of results at all path nodes: `prpath, s_vmis, u_y`.

4.5 Identification and Fixing of Model Errors

Before submitting a finite element model deck to the solver engine (`solve`), automated database verification and topological cleanup should be performed to prevent numerical ill-conditioning, rigid body modes, or matrix singularities. uFEM includes dedicated diagnostic verification and topology manipulation commands:

4.5.1 Automated Database Consistency Test (`datatest`)

The `datatest` command performs an exhaustive integrity check across the active finite element database:

- Identifies orphaned nodes (nodes defined in memory but unreferenced by any finite element).
- Identifies elements assigned invalid or missing material IDs (`mat`), real constant set IDs (`rs`), or element type IDs (`et`).
- Identifies degenerate elements (e.g. quad elements collapse to zero area or solid hex elements with zero volume).
- Identifies zero-length line entities and un-meshed geometric keypoints.

The solver caller automatically runs `datatest` before exporting database files (`.fem`).

4.5.2 Mesh Quality and Geometric Distortion Checks (`checkmesh` and `echeck`)

Distorted element shapes can severely impair stiffness matrix condition numbers:

- `checkmesh[, max_aspect, min_jac]`: Scans all selected elements for aspect ratio limit `max_aspect` (default 5.0) and minimum Jacobian determinant threshold `min_jac` (default 0.0). Automatically selects distorted elements exceeding these limits, enabling immediate visual inspection with `eplot`.
- `echeck`: Evaluates detailed elemental shape metrics (taper, skewness, minimum interior angle, warping factor) and reports warning/error summaries to output windows.

4.5.3 Geometric Assembly Repairs: Boolean Joining (`gejoin`) and Division Sync (`gesync`)

Geometric alignment errors between adjacent entities must be resolved prior to meshing:

- `gejoin[, ent1, ent2]`: Merges coincident keypoints and shared boundary edges between adjacent geometry entities (`ent1` and `ent2`, or all entities if parameters are omitted). Performs a geometric Boolean Union across shared interface boundaries, preventing unjoined geometry gaps.
- `gesync`: Automatically synchronizes line division counts (`gediv`) across coincident contact boundary edges between adjacent geometry entities. Guarantees 100% conformal node seeding across geometry interfaces when meshing.

4.5.4 Keypoint Manipulation: Merging (`kmerge`) and Splitting (`ksplit`)

Geometric keypoint topology can be repaired or modified prior to meshing:

- `kmerge[, tol]`: Scans keypoints and merges coincident keypoints lying within distance tolerance `tol` (default 10^{-6} m) into a single keypoint ID. Essential after importing CSG geometry blocks to enforce shared interface boundaries.
- `ksplit, from[, to]`: Splits keypoint ID `from` into duplicate independent keypoints, disconnecting adjacent geometric lines or volumes sharing that keypoint.

4.5.5 Nodal Mesh Manipulation: Merging (`nmerge`) and Splitting (`nsplit`)

Finite element nodal topology can be repaired or modified after meshing:

- `nmerge[, tol]`: Merges duplicate coincident nodes within spatial tolerance `tol` into a single node ID. Re-indexes element node connectivity tables, ensuring continuous displacement field transfer across meshed volume interfaces.
- `nsplit, from[, to]`: Splits node ID `from` into duplicate independent node IDs. Disconnects element connectivity at that location to model physical cracks, structural hinges, or expansion joints.

4.6 Local AI Assistant Integration (`-z`, `-zp`, `ai`)

> **Note (Experimental Feature):** The Local AI Assistant integration is an **experimental feature** under active development. Output generated by LLMs must be verified by structural engineers before critical application.

`uFEM` includes native integration with local Large Language Models (LLMs) via `llama.cpp` (`llama-server`) operating over C89 HTTP REST sockets.

4.6.1 Startup Configuration

- `-z`: Enables the Local AI assistant integration at program launch.
- `-zp host:port`: Specifies the local AI server address and port (default: `127.0.0.1:8080`).
- `-zg`: Enables Google Gemini Cloud AI integration with automatic local LLM backup.
- `-zgk API_KEY`: Specifies the Google Gemini API key or access token.

4.6.2 Interactive AI Commands

- `ai, <prompt>`: Sends a natural language description to the AI assistant, generates uFEM macro commands, and opens the Review Window.
- `ai,gen, <prompt>`: Generates and displays the uFEM macro commands without auto-executing.

- ai,geom, <prompt>: Generates 2D/3D CAD geometry entities (gerect, gecv2d, gebox, gecv3d) and automatic meshing.
- ai,explain, <cmd_or_file>: Explains uFEM command line syntax or input .mac macro files line by line for structural engineers.
- ai,ask, <question>: Answers targeted natural language questions about active database state in memory.
- ai,fix: Analyzes database status and diagnostic warnings (datatest) and queries the AI assistant for automated command fixes.
- ai,google,key,YOUR_KEY: Configures your Google Gemini API Key dynamically during an active tfem or gfem session.
- ai,auth (or ai,login): Launches your system browser for 1-click Google AI Studio API key management or setting local AI endpoints interactively.
- ai,status: Verifies connectivity and displays local and cloud AI server status.

4.6.3 GTK+ Desktop & WebAssembly GUI AI Review Window

In gfem (GTK+ Desktop GUI) and gfem_web (Web GUI / WebAssembly), generating macro scripts opens the interactive **uFEM AI Macro Editor & Verification** window featuring:

- **Syntax Highlighting**: Monospace code rendering with green comments (! ...), blue keywords, and red numerical parameters.
- **Execute Script**: Runs the reviewed macro in active memory.
- **Verify Syntax**: Non-destructive pre-execution validation of command names and parameter bounds.
- **Clean Comments**: Strips comments and blank lines to produce clean, compact uFEM code.
- **Save Macro (.mac)**: Exports the script to a reusable .mac file.
- **WASM Gemini Integration**: In client-side WebAssembly browser mode, queries Google Gemini API directly via native browser fetch() without requiring a local server process.

All prompt interactions, generated macros, and reports are automatically logged to fem_ai.log.

4.6.4 Practical AI Workflow Examples

Example 1: Generating a 2D Cantilever Beam Macro To generate a structural FEA model setup from natural language in tfem or gfem:

```
uFEM > ai, Create a 2D cantilever beam 10m long fixed at left end with 50kN point load
[I] Querying local AI server at http://127.0.0.1:8080..

--- Local AI Assistant Output ---
! Material and Element Definition
et,1,2
```

```
mat,1,210e9,0.3
k,1,0,0
k,2,10,0
l,1,1,2
ldiv,1,10
mesh,1
d,1,all,0
f,2,fy,-50000
solve
--------------------------------
[I] Executing AI-generated macro script..
```

Example 2: Automated Technical Model Verification Report To generate a comprehensive structural audit report from active database memory:

```
uFEM > ai,report
[I] Querying local AI server at http://127.0.0.1:8080..

--- Local AI Assistant Output ---
========================================================================
          uFEM Finite Element Analysis - AI Model & Verification Report
========================================================================
1. Model Overview: 2D Plane Frame Model with 150 Nodes and 120 Elements.
2. Materials: Material 1 (E=210 GPa, nu=0.3).
3. Boundary Conditions: Fixed Support at Node 1 (Ux=Uy=0); Point Load Fy=-50kN at Node 2.
4. Structural Sanity: Model database is complete; no unassigned properties detected.
------------------------------------------------------------------------
[I] Technical AI Model Report saved to file 'fem_ai_report.txt'
```

Example 3: Automated Error Diagnosis and Model Fixing If solver assembly fails or database consistency warnings are raised:

```
uFEM > datatest
[W] Warning: Keypoints 3 and 4 are coincident within 1e-5 tolerance!
uFEM > ai,fix
[I] Querying local AI server for model repair..
[I] Suggested Repair: kmerge, 1e-5; mesh, 1
```

5 tfem and toutview - Terminal Applications

`tfem` provides the full interactive command interpreter of `gfem` within a terminal window (VT100 ANSI escape sequences). It operates without requiring an X11 display or graphics adapter, making it ideal for remote SSH sessions, batch execution scripts, and high-performance compute clusters.

Limitations: The main limitation of this interface is unavailability on any graphics, including outputs. Text outputs ar working here (`pr..` commands are redirected to the program defined by the `outview,PROGRAM` command).

The `help,..` command works here.

The `toutview` is a text-only version of the `goutview`. If configured, is called for `pr..` commands outputs.

6 fem - Standalone Solver Engine

The solver binary `fem` contains the core numerical linear algebra routines, element stiffness matrix assembly, skyline and Preconditioned Conjugate Gradient (PCG) equation solvers, and non-linear iteration loops [?, ?].

6.1 Mandatory Executable Flag (-p) and File Specification

When executing directly from the command line, the solver **MUST ALWAYS** be invoked with the `-p` parameter:

```
fem -p jobname
```

By default, `fem -p jobname` reads pre-parsed binary mesh data from `jobname.fem` and writes primary binary results to `jobname.res`.

Explicit input and output file paths can be specified using the `-i` and `-o` parameters:

```
fem -p -i /path/to/input.fem -o /path/to/output.res
```

6.2 Command Line Options Reference

The `fem` solver supports the following command line switches:

- `-p`: Mandatory flag indicating pre-parsed input file format exported by `gfem`/`tfem`.
- `-i FILE`: Specifies canonical path to input `.fem` model file.
- `-o FILE`: Specifies canonical path to output `.res` binary results file.
- `-t N`: Sets the number of parallel worker threads (N) for stiffness assembly and solver execution (pthreads build).
- `-fbc`: Enables fast boundary condition assembly (excludes zero displacement DOFs from matrix system, skips computations of reactions; it's advisable for subspace modelling problems).
- `-x`: Suppresses the interactive OpenGL/X11 convergence visualization window during non-linear solver iterations (used by default in batch and regression test runs).
- `-ti FILE`: Specifies input thermal result file containing nodal temperatures $T(x, y, z)$ for coupled thermal-structural analysis.
- `-to FILE`: Specifies output file for thermal analysis temperature fields.
- `-ao FILE`: Specifies alternative output filename (e.g. `-ao model.vtk`).
- `-at TYPE`: Specifies alternative output file format type (`0` = plain ANSI text, `1` = VTK legacy ASCII format).

6.3 VTK Results Export (-ao and -at 1) & Format Limitations

uFEM supports exporting computed results directly to VTK (Visualization Toolkit Unstructured Grid ASCII format) for visualization in external post-processors such as ParaView or MayaVi.

```
fem -p -i model.fem -o model.res -ao model.vtk -at 1
```

Capabilities & Output Quantities: The VTK export writes an ASCII text with the header type `DATASET UNSTRUCTURED_GRID` containing:

- Nodal coordinates (`POINTS`) and cell connectivity topology (`CELLS`).
- Nodal displacement field vectors: `u_x`, `u_y`, `u_z`.
- Strain components: `e_x`, `e_y`, `e_z`.
- Stress tensor components: `s_x`, `s_y`, `s_z`, `s_xy`, `s_yz`, `s_zx`.
- Principal stresses: `s_1`, `s_2`, `s_3`.

6.4 Vector Graphics Export (`export,ps` and `export,svg`)

uFEM includes built-in headless vector graphics export capability for high-quality publication diagrams in PostScript (`.ps`) and Scalable Vector Graphics (`.svg`) formats.

```
ples,s_x
export,ps,output.ps
export,svg,output.svg
```

Features & Layout Capabilities:

- **Color Contour Rendering**: Automatically extracts element stress/result distributions (`ples`) and fills finite element polygons with continuous RGB palette spectrum color mapping.
- **5-Tick Numeric Colorbar Legend**: Renders a vertical colorbar with 5 numerical tick mark values (ranging from minimum to maximum result field values) on the right side of the document.
- **Viewport Isolation**: Automatically calculates Model View bounding boxes and reserves a 140px right margin so the colorbar legend box never obscures or overlaps the model geometry or stress results.

7 Benchmark Macro Examples

7.1 Example 1: 2D Simply Supported Beam under Uniform Distributed Load

7.1.1 Structural Schematic & Internal Force Diagrams

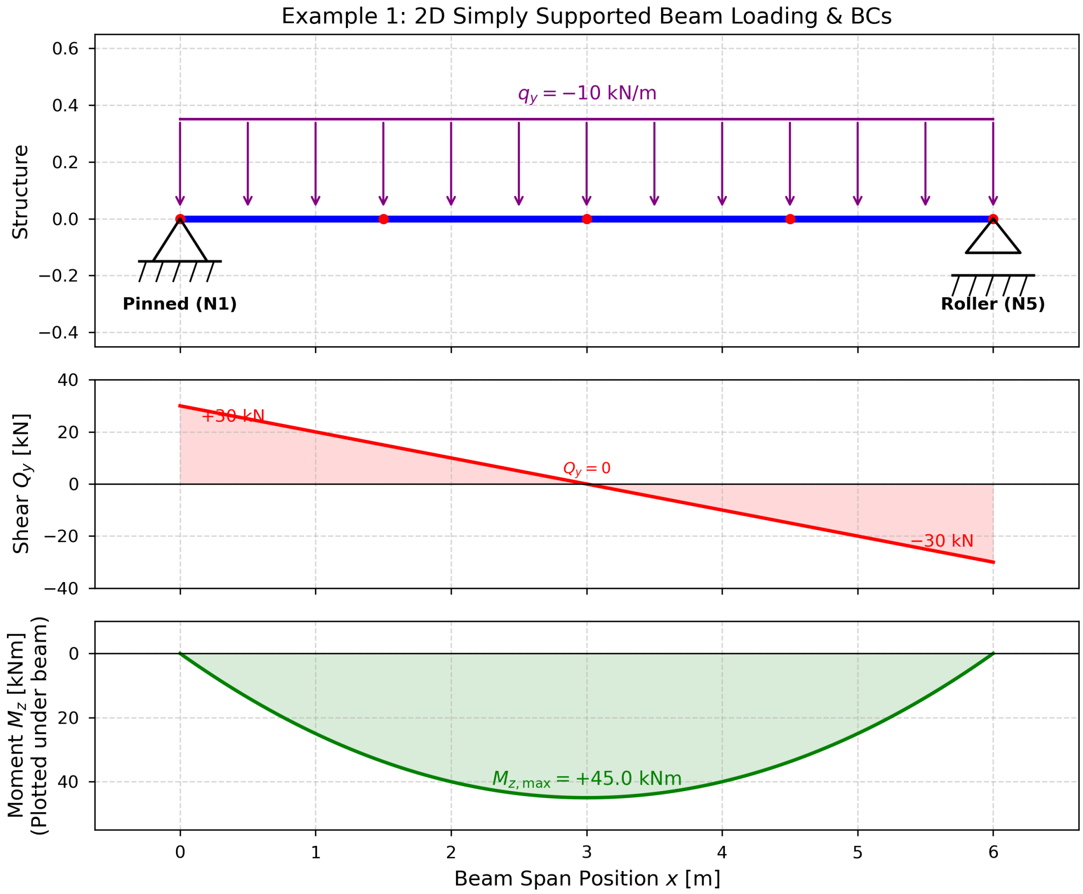


Figure 2: Structural Schematic, Shear Force (Q_y), and Bending Moment (M_z) Diagrams for Example 1.

7.1.2 Macro Source Code Listing (example_beam2d.mac)

```
cleardata
jobname,example_beam2d

! Material 1: Structural Steel S235
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3
mp,dens,1,7850

! Element Type 1: Beam Element (Type 14)
et,1,14
cs,IPE 300,1,14

! Nodes (Span L = 6m, 4 elements)
n,1,0.0,0.0,0.0
n,2,1.5,0.0,0.0
n,3,3.0,0.0,0.0
```

```
n,4,4.5,0.0,0.0
n,5,6.0,0.0,0.0

! Elements
edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Conditions: Left Pinned (N1), Right Roller (N5)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0
d,5,uy,0
d,5,uz,0

! Continuous Distributed Load qy = -10 kN/m on Elements 1 to 4
el,1,qy,-10000
el,2,qy,-10000
el,3,qy,-10000
el,4,qy,-10000


! Solve static linear system
allsel
solve,linear,5,100

! Output Plots & Capture Images
plotp,wireonly,1
ples,m_z
gl2bmp,example_beam2d_mz.bmp
ples,f_y
gl2bmp,example_beam2d_qy.bmp

exit
```

Listing 2: Complete verified macro example_beam2d.mac

7.2 Example 2: 3D Portal Frame under Lateral Point Loads & Gravity

7.2.1 Structural Schematic

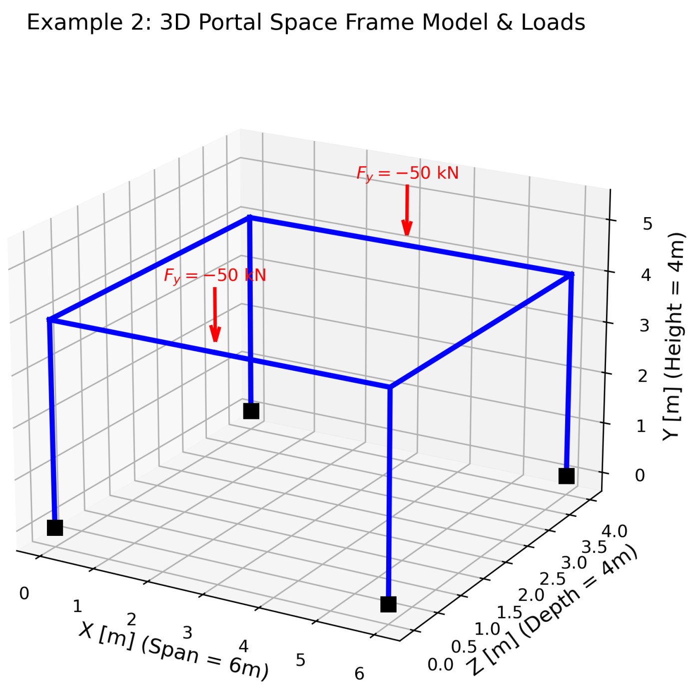


Figure 3: 3D Isometric Geometry, Nodal Loading, and Support Constraints for Example 2.

7.2.2 Macro Source Code Listing (example_frame3d.mac)

```
cleardata
jobname,example_frame3d

! Material: Steel S355
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3
mp,dens,1,7850

! Element Type: 3D Frame Element (Type 14)
et,1,14
cs,HEA 200,1,14

! Nodes (3D Portal Frame: 6m wide, 4m high, 4m deep)
! Ground Nodes (Y = 0)
n,1,0.0,0.0,0.0
n,2,6.0,0.0,0.0
n,3,6.0,0.0,4.0
n,4,0.0,0.0,4.0

! Roof Nodes (Y = 4m)
n,5,0.0,4.0,0.0
n,6,6.0,4.0,0.0
n,7,6.0,4.0,4.0
n,8,0.0,4.0,4.0

```

```
! Mid-span Beam Nodes (Y = 4m)
n,9,3.0,4.0,0.0
n,10,3.0,4.0,4.0
n,11,0.0,4.0,2.0
n,12,6.0,4.0,2.0

! Elements (Type 14)
edef,1,1,1,1
! 4 Vertical Columns
e,1,1,5
e,2,2,6
e,3,3,7
e,4,4,8

! Roof Girders (X-direction)
e,5,5,9
e,6,9,6
e,7,8,10
e,8,10,7

! Roof Beams (Z-direction)
e,9,5,11
e,10,11,8
e,11,6,12
e,12,12,7

! Boundary Conditions: Fixed Supports at Ground (N1, N2, N3, N4)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0
d,1,roty,0
d,1,rotz,0

d,2,ux,0
d,2,uy,0
d,2,uz,0
d,2,rotx,0
d,2,roty,0
d,2,rotz,0

d,3,ux,0
d,3,uy,0
d,3,uz,0
d,3,rotx,0
d,3,roty,0
d,3,rotz,0

d,4,ux,0
d,4,uy,0
d,4,uz,0
d,4,rotx,0
d,4,roty,0
d,4,rotz,0

! Point Loads on Roof Mid-spans (N9, N10: Fy = -50 kN)
f,9,fy,-50000
f,10,fy,-50000

! Gravity Load on Structure
accel,y,-9.81

! Solve static linear system
```

```
allsel
solve,linear,5,100

! Display Deformed Shape & Internal Forces
plotp,wireonly,1
ples,m_z
gl2bmp,example_frame3d_mz.bmp
ples,f_y
gl2bmp,example_frame3d_qy.bmp

exit
```

Listing 3: Complete verified macro `example_frame3d.mac`

7.3 Example 3: Plane Stress Wall with Circular Hole

7.3.1 Structural Schematic

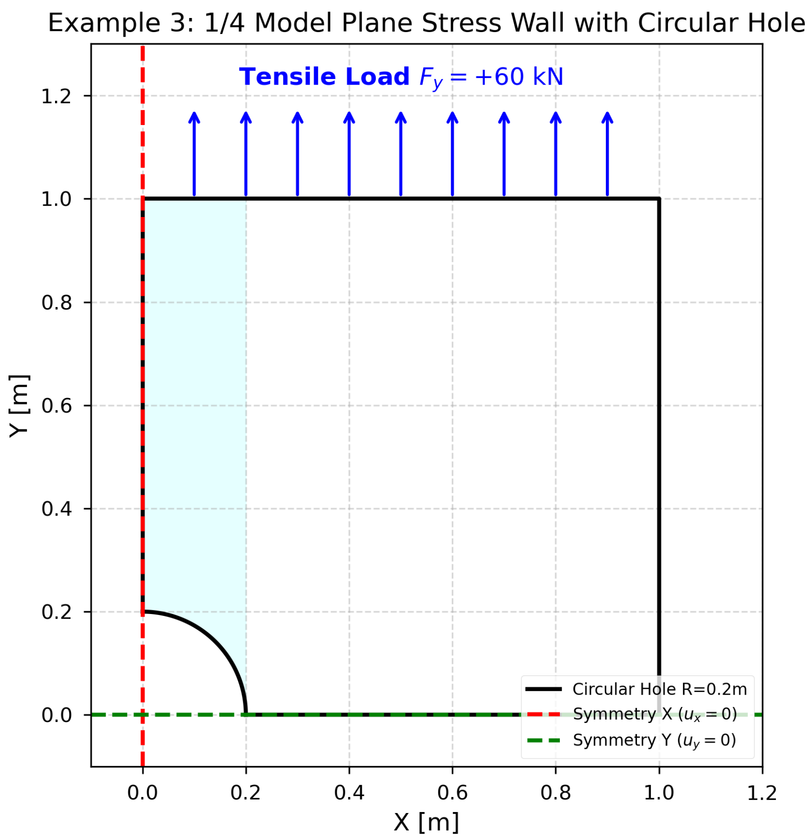


Figure 4: 1/4 Symmetry Model Geometry, Boundary Conditions, and Tensile Load for Example 3.

7.3.2 Macro Source Code Listing (`example_wall_hole.mac`)

```
cleardata
jobname,example_wall_hole

! Material: Steel S235 Elastic-Plastic
mat,1,4
mp,ex,1,210e9
mp,prxy,1,0.3
mp,fyc,1,235e6
mp,hard,1,1e9

! Element Type: 2D Plane Stress (Type 2, thickness t = 10mm)
et,1,2
rs,1,2
r,height,1,0.01

! Geometry: 1/4 Symmetry Model of 1.0m x 1.0m Plate with Hole (R = 0.2m)
k,1,0.2,0.0
k,2,0.1414,0.1414
k,3,0.0,0.2

k,4,1.0,0.0
```

```
k,5,1.0,1.0
k,6,0.0,1.0

k,7,1.0,0.2
k,8,0.2,1.0

a,1,1,4,7,2
a,2,2,7,5,8
a,3,3,2,8,6

! Mesh Division (8x8 elements per area)
gediv,1,8,8
gediv,2,8,8
gediv,3,8,8
mesh

! Symmetry Boundary Conditions
nsel,s,loc,x,0
d,all,ux,0

nsel,s,loc,y,0
d,all,uy,0

! Tensile Load at Top Edge (Y = 1.0m)
nsel,s,loc,y,1
f,all,fy,60000

! Solve static linear & non-linear steps
allsel
solve,linear,5,100

! Plot Von Mises Stress Field
plns,s_vmis
gl2bmp,example_wall_hole_vmis.bmp
plns,s_x
gl2bmp,example_wall_hole_sx.bmp

exit
```

Listing 4: Complete verified macro `example_wall_hole.mac`

7.4 Example 4: Cooling Tower Shell Structure with Frame Columns

7.4.1 Structural Schematic

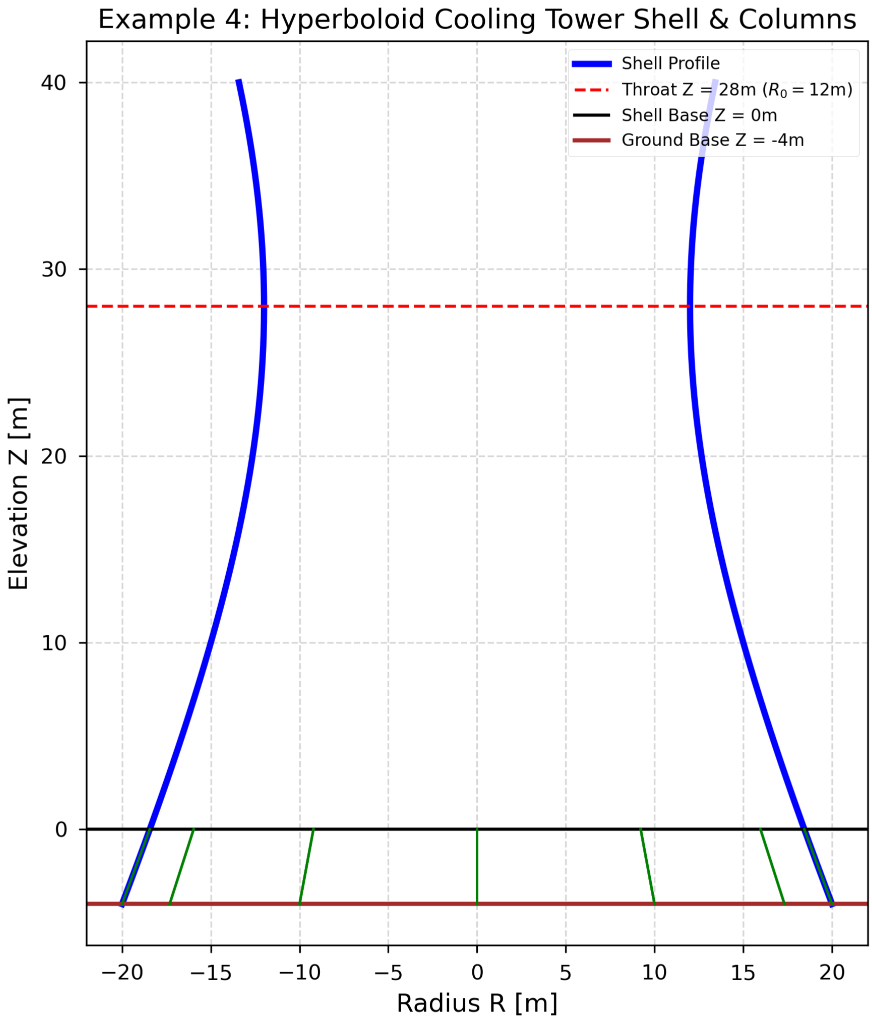


Figure 5: Hyperboloid Profile Geometry, Throat Elevation, and V-Leg Supporting Columns for Example 4.

7.4.2 Macro Source Code Listing (example_tower_shell.mac)

```
cleardata
jobname,example_tower_shell

! Material: Reinforced Concrete C30/37
mat,1,1
mp,ex,1,33e9
mp,prxy,1,0.2
mp,dens,1,2500

! Element Types: 1=Flat Shell (Type 24), 2=Frame Column (Type 14)
et,1,24
rs,1,24
r,height,1,0.18

```

```
et,2,14
rs,2,14
r,area,2,0.0625
r,ix,2,3.25e-4
r,iy,2,3.25e-4
r,iz,2,3.25e-4

script,geom

! Parameters for Hyperboloid Cooling Tower (Height H = 40m, Throat at Z = 28m)
eval,zthroat,28.0,*,1.0
eval,C,24.0,*,1.0
eval,R0,12.0,*,1.0

! Generate Base Support Nodes (Z = -4m)
for,it,0,11,
  eval,ang,it,*,30.0
  eval,temp1,0.0,-,zthroat
  eval,temp2,temp1,/,C
  eval,temp3,temp2,^,2
  eval,temp4,temp3,+,1.0
  eval,temp5,temp4,^,0.5
  eval,R,temp5,*,R0
  eval,cx,R,c,ang
  eval,cy,R,s,ang
  eval,cz,-4.0,*,1.0
  ieval,nidx,it,+,1
  n,nidx,cx,cy,cz
endfor

! Generate Hyperbolic Shell Nodes (Z = 0 to 40m)
for,iz,0,8,
  for,it,0,11,
    eval,ang,it,*,30.0
    eval,z,iz,*,5.0
    eval,temp1,z,-,zthroat
    eval,temp2,temp1,/,C
    eval,temp3,temp2,^,2
    eval,temp4,temp3,+,1.0
    eval,temp5,temp4,^,0.5
    eval,R,temp5,*,R0
    eval,cx,R,c,ang
    eval,cy,R,s,ang
    ieval,nidx,iz,*,100
    ieval,nidx,nidx,+,it
    ieval,nidx,nidx,+,101
    n,nidx,cx,cy,z
  endfor
endfor

! V-Leg Supporting Frame Columns (Type 14)
edef,2,2,1
for,it,0,11,
  ieval,temp,it,+,1
  ieval,q,temp,/,12
  ieval,m,q,*,12
  ieval,next_it,temp,-,m

  ieval,n_gnd1,it,+,1
  ieval,n_gnd2,next_it,+,1
  ieval,n_bot1,it,+,101
  ieval,n_bot2,next_it,+,101

```

```
  ieval,eidx1,it,*,2
  ieval,eidx1,eidx1,+,501
  e,eidx1,n_gnd1,n_bot2

  ieval,eidx2,it,*,2
  ieval,eidx2,eidx2,+,502
  e,eidx2,n_gnd2,n_bot1
endfor

! Quad Shell Elements (Type 24)
edef,1,1,1
for,iz,0,7,
  for,it,0,11,
    ieval,temp,it,+,1
    ieval,q,temp,/,12
    ieval,m,q,*,12
    ieval,next_it,temp,-,m

    ieval,n1,iz,*,100
    ieval,n1,n1,+,it
    ieval,n1,n1,+,101

    ieval,n2,iz,*,100
    ieval,n2,n2,+,next_it
    ieval,n2,n2,+,101

    ieval,n3,iz,+,1
    ieval,n3,n3,*,100
    ieval,n3,n3,+,next_it
    ieval,n3,n3,+,101

    ieval,n4,iz,+,1
    ieval,n4,n4,*,100
    ieval,n4,n4,+,it
    ieval,n4,n4,+,101

    ieval,eidx,iz,*,12
    ieval,eidx,eidx,+,it
    ieval,eidx,eidx,+,1

    e,eidx,n1,n2,n3,n4
  endfor
endfor

! Fixed Ground Boundary Conditions
nsel,s,loc,z,-4
d,all,ux,0
d,all,uy,0
d,all,uz,0
d,all,rotx,0
d,all,roty,0
d,all,rotz,0

! Self-weight Gravity Load
allsel
accel,z,-9.81

! Solve static linear system
allsel
solve,linear,5,100

! Separate Analysis: Shell Results vs Supporting Columns
esel,s,type,1
```

```
plns,s_vmis
gl2bmp,example_tower_shell_stress.bmp

esel,s,type,2
ples,m_z
gl2bmp,example_tower_columns_mz.bmp

exit
```

Listing 5: Complete verified macro `example_tower_shell.mac`

7.5 Example 5: Cooling Tower 3D Solid Hexahedral Structure

7.5.1 Structural Schematic & 3D Section Cut

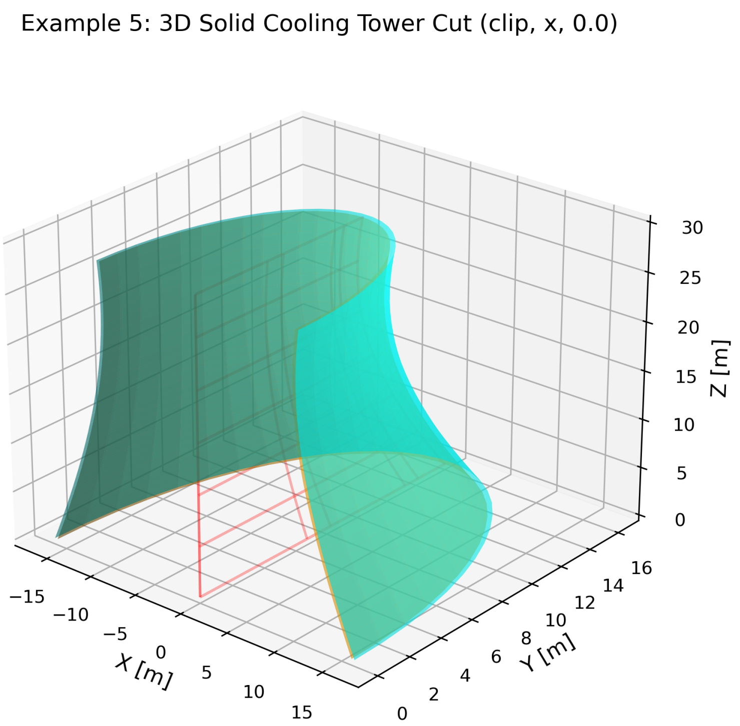


Figure 6: 3D Hexahedral Solid Mesh Geometry with Hardware OpenGL Cut Plane (`clip, x, 0.0`) for Example 5.

7.5.2 Macro Source Code Listing (`example_tower_solid.mac`)

```
cleardata
jobname,example_tower_solid

! Material: Reinforced Concrete C30/37
mat,1,1
mp,ex,1,33e9
mp,prxy,1,0.2
mp,dens,1,2500

! Element Type: 8-Node 3D Hexahedral Solid Brick (Type 9)
et,1,9
rs,1,9

script,geom

! Parameters for Solid Cooling Tower Model
eval,zthroat,25.0,*,1.0
eval,C,20.0,*,1.0
eval,R0,10.0,*,1.0
eval,t_shell,0.3,*,1.0

! Generate Inner and Outer Shell Solid Nodes (Radially extruded 3D Solid Hexahedra)
for,iz,0,6,
```

```
  for,it,0,11,
    eval,ang,it,*,30.0
    eval,z,iz,*,5.0
    eval,temp1,z,-,zthroat
    eval,temp2,temp1,/,C
    eval,temp3,temp2,^,2
    eval,temp4,temp3,+,1.0
    eval,temp5,temp4,^,0.5
    eval,R_in,temp5,*,R0
    eval,R_out,R_in,+,t_shell

    ! Inner surface node
    eval,cx_in,R_in,c,ang
    eval,cy_in,R_in,s,ang
    ieval,nidx_in,iz,*,100
    ieval,nidx_in,nidx_in,+,it
    ieval,nidx_in,nidx_in,+,101
    n,nidx_in,cx_in,cy_in,z

    ! Outer surface node
    eval,cx_out,R_out,c,ang
    eval,cy_out,R_out,s,ang
    ieval,nidx_out,iz,*,100
    ieval,nidx_out,nidx_out,+,it
    ieval,nidx_out,nidx_out,+,501
    n,nidx_out,cx_out,cy_out,z
  endfor
endfor

! Generate 3D Solid Hexahedral Brick Elements (Type 9)
edef,1,1,1
for,iz,0,5,
  for,it,0,11,
    ieval,temp,it,+,1
    ieval,q,temp,/,12
    ieval,m,q,*,12
    ieval,next_it,temp,-,m

    ! Inner bottom nodes
    ieval,n1,iz,*,100
    ieval,n1,n1,+,it
    ieval,n1,n1,+,101

    ieval,n2,iz,*,100
    ieval,n2,n2,+,next_it
    ieval,n2,n2,+,101

    ! Inner top nodes
    ieval,n3,iz,+,1
    ieval,n3,n3,*,100
    ieval,n3,n3,+,next_it
    ieval,n3,n3,+,101

    ieval,n4,iz,+,1
    ieval,n4,n4,*,100
    ieval,n4,n4,+,it
    ieval,n4,n4,+,101

    ! Outer bottom nodes
    ieval,n5,iz,*,100
    ieval,n5,n5,+,it
    ieval,n5,n5,+,501

```

```
    ieval,n6,iz,*,100
    ieval,n6,n6,+,next_it
    ieval,n6,n6,+,501

    ! Outer top nodes
    ieval,n7,iz,+,1
    ieval,n7,n7,*,100
    ieval,n7,n7,+,next_it
    ieval,n7,n7,+,501

    ieval,n8,iz,+,1
    ieval,n8,n8,*,100
    ieval,n8,n8,+,it
    ieval,n8,n8,+,501

    ieval,eidx,iz,*,12
    ieval,eidx,eidx,+,it
    ieval,eidx,eidx,+,1

    e,eidx,n1,n2,n3,n4,n5,n6,n7,n8
  endfor
endfor

! Fixed Ground Base Boundary Conditions (Z = 0)
nsel,s,loc,z,0
d,all,ux,0
d,all,uy,0
d,all,uz,0

! Self-weight Gravity Load
allsel
accel,z,-9.81

! Solve static linear system
allsel
solve,linear,5,100

! Plot 3D Solid Stress Field
plns,s_vmis
gl2bmp,example_tower_solid_stress.bmp

! Apply 3D Clipping Section Cut at X = 0 to reveal interior solid stress
clip,x,0.0
plns,s_vmis
gl2bmp,example_tower_solid_cut.bmp
clip,off

exit
```

Listing 6: Complete verified macro example_tower_solid.mac

7.6 Example 6: 2D Block Friction Contact (Element 21 - 2D Link Contact)

7.6.1 Structural Schematic

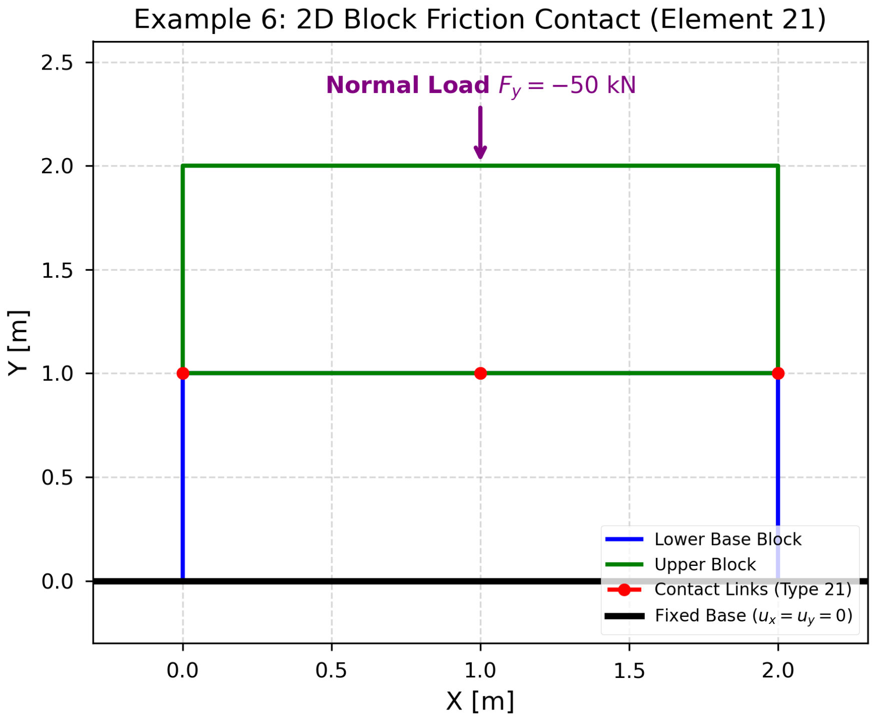


Figure 7: 2D Multi-Body Block Compression & Friction Contact Model for Example 6.

7.6.2 Macro Source Code Listing (`example_contact2d_blocks.mac`)

```
cleardata
jobname,example_contact2d_blocks

! Material 1: Lower Base Block (Elastic-Plastic Steel)
mat,1,4
mp,ex,1,210e9
mp,prxy,1,0.3
mp,fyc,1,235e6
mp,hard,1,1e9

! Element Types: 1=2D Plane Stress (Type 2), 2=2D Contact Link (Type 21)
et,1,2
rs,1,2
r,height,1,0.02

et,2,21
rs,2,21
r,27,2,0.3      ! Friction coefficient mu = 0.3
r,28,2,1        ! Contact pair ID
r,29,2,1e11     ! Normal penalty stiffness kn = 100 GPa/m
r,30,2,1e10     ! Tangential penalty stiffness kt = 10 GPa/m

! Nodes for Lower Block (0 <= X <= 2.0, 0 <= Y <= 1.0)
n,1,0.0,0.0
n,2,1.0,0.0
n,3,2.0,0.0
```

```
n,4,2.0,1.0
n,5,1.0,1.0
n,6,0.0,1.0

! Nodes for Upper Block (0 <= X <= 2.0, 1.0 <= Y <= 2.0) - Initial touching
    interface at Y = 1.0
n,11,0.0,1.0
n,12,1.0,1.0
n,13,2.0,1.0
n,14,2.0,2.0
n,15,1.0,2.0
n,16,0.0,2.0

! Elements for Lower Block (Type 2)
edef,1,1,1
e,1,1,2,5,6
e,2,2,3,4,5

! Elements for Upper Block (Type 2)
e,3,11,12,15,16
e,4,12,13,14,15

! 2D Contact Link Elements (Type 21) across interfacial node pairs (N6-N11, N5-
    N12, N4-N13)
edef,2,2,1
e,5,6,11
e,6,5,12
e,7,4,13

! Boundary Conditions: Fix bottom edge of lower block (Y = 0)
nsel,s,loc,y,0
d,all,ux,0
d,all,uy,0

! Compressive Vertical Load on Top Block (Y = 2.0)
allsel
nsel,s,loc,y,2.0
f,all,fy,-50000

! Solve Non-linear Contact & Plasticity System
allsel
solve,linear,5,100

! Plot Stress Contours
plns,s_vmis
gl2bmp,example_contact2d_vmis.bmp

exit
```

Listing 7: Complete verified macro `example_contact2d_blocks.mac`

7.7 Example 7: 3D Plate Contact Interface (Element 22 - 3D Link Contact)

7.7.1 Structural Schematic

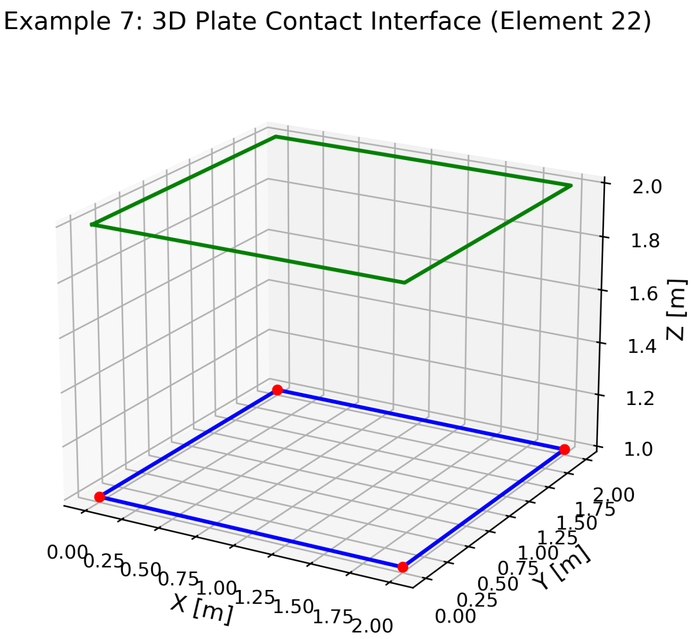


Figure 8: 3D Double-Plate Compression Contact Interface Model for Example 7.

7.7.2 Macro Source Code Listing (`example_contact3d_plates.mac`)

```
cleardata
jobname,example_contact3d_plates

! Material 1: Structural Steel S355
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3
mp,dens,1,7850

! Element Types: 1=3D Hex Solid (Type 9), 2=3D Contact Link (Type 22)
et,1,9
rs,1,9

et,2,22
rs,2,22
r,27,2,0.25    ! Friction coefficient mu = 0.25
r,28,2,1       ! Pair ID
r,29,2,1e11    ! Normal stiffness kn = 100 GPa/m
r,30,2,1e10    ! Tangential stiffness kt = 10 GPa/m

! Nodes for Lower Plate (0 <= Z <= 1.0)
n,1,0,0,0
n,2,2,0,0
n,3,2,2,0
n,4,0,2,0
n,5,0,0,1
n,6,2,0,1
n,7,2,2,1
```

```
n,8,0,2,1

! Nodes for Upper Plate (1.0 <= Z <= 2.0)
n,11,0,0,1
n,12,2,0,1
n,13,2,2,1
n,14,0,2,1
n,15,0,0,2
n,16,2,0,2
n,17,2,2,2
n,18,0,2,2

! Elements (Type 9)
edef,1,1,1
e,1,1,2,3,4,5,6,7,8
e,2,11,12,13,14,15,16,17,18

! 3D Contact Link Elements (Type 22) between N5-N11, N6-N12, N7-N13, N8-N14
edef,2,2,1
e,3,5,11
e,4,6,12
e,5,7,13
e,6,8,14

! Boundary Conditions: Fix base of lower plate (N1, N2, N3, N4 at Z = 0)
d,1,ux,0
d,1,uy,0
d,1,uz,0

d,2,ux,0
d,2,uy,0
d,2,uz,0

d,3,ux,0
d,3,uy,0
d,3,uz,0

d,4,ux,0
d,4,uy,0
d,4,uz,0

! Compressive Vertical Load on Top Plate (N15, N16, N17, N18 at Z = 2.0)
f,15,fz,-20000
f,16,fz,-20000
f,17,fz,-20000
f,18,fz,-20000

! Solve static linear system
allsel
solve,linear,5,100

! Plot Stress Contours
plns,s_vmis
gl2bmp,example_contact3d_vmis.bmp

exit
```

Listing 8: Complete verified macro `example_contact3d_plates.mac`

7.8 Example 8: 3D Sheet Metal Clinching (Element 23 - 3D Node-to-Segment NTS Contact)

7.8.1 Structural Schematic

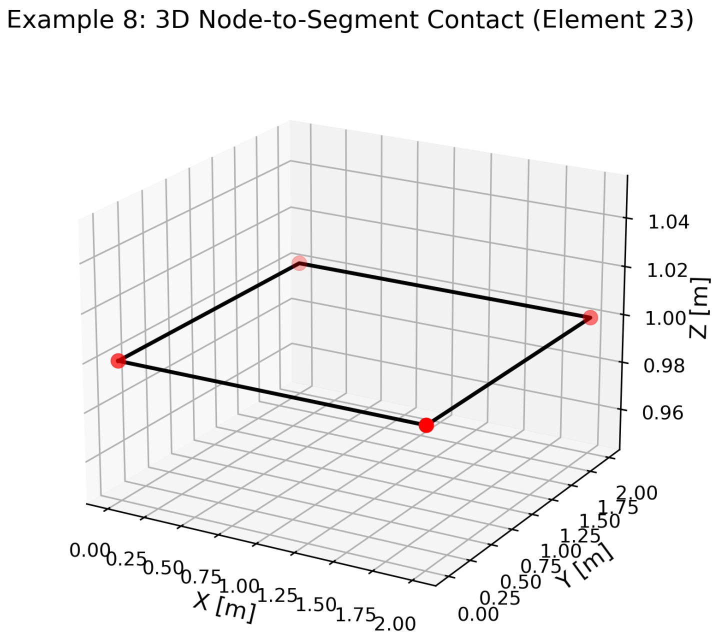


Figure 9: 3D Node-to-Segment (NTS) Surface Penalty Contact Model for Example 8.

7.8.2 Macro Source Code Listing (example_contact3d_nts_clinch.mac)

```
cleardata
jobname,example_contact3d_nts_clinch

! Material 1: S235 Structural Steel Sheet Metal
et,1,9
rs,1,9
mat,1,4
mp,ex,1,210e9
mp,prxy,1,0.3
mp,fyc,1,235e6
mp,hard,1000e6,1

! Nodes for Lower Sheet Metal (0 <= Z <= 1.0)
n,1,0,0,0
n,2,2,0,0
n,3,2,2,0
n,4,0,2,0
n,5,0,0,1
n,6,2,0,1
n,7,2,2,1
n,8,0,2,1

! Nodes for Upper Sheet Metal (1.0 <= Z <= 2.0)
n,11,0,0,1
n,12,2,0,1
n,13,2,2,1
n,14,0,2,1
```

```
n,15,0,0,2
n,16,2,0,2
n,17,2,2,2
n,18,0,2,2

! Solid Hexahedral Bricks (Type 9)
edef,1,1,1
e,1,1,2,3,4,5,6,7,8
e,2,11,12,13,14,15,16,17,18

! 3D Node-to-Segment (NTS) Contact Elements (Type 23)
! Slave nodes N11, N12, N13, N14 vs Master surface quad segment (N5, N6, N7, N8)
et,2,23
rs,2,23
r,27,2,0.2     ! Friction coefficient mu = 0.2
r,28,2,1       ! Pair ID
r,29,2,1e11    ! Normal stiffness kn = 100 GPa/m
r,30,2,1e10    ! Tangential stiffness kt = 10 GPa/m

edef,2,2,1
e,3,11,5,6,7,8
e,4,12,5,6,7,8
e,5,13,5,6,7,8
e,6,14,5,6,7,8

! Boundary Conditions: Fix base of lower sheet metal (Z = 0)
d,1,ux,0
d,1,uy,0
d,1,uz,0

d,2,ux,0
d,2,uy,0
d,2,uz,0

d,3,ux,0
d,3,uy,0
d,3,uz,0

d,4,ux,0
d,4,uy,0
d,4,uz,0

! Compressive Load on Top Sheet (Z = 2.0)
f,15,fz,-25000
f,16,fz,-25000
f,17,fz,-25000
f,18,fz,-25000

! Solve Non-linear NTS Surface Contact
allsel
solve,linear,5,100

! Plot Von Mises Stress Contour Field
plns,s_vmis
gl2bmp,example_contact_nts_vmis.bmp

exit
```

Listing 9: Complete verified macro `example_contact3d_nts_clinch.mac`

7.9 Example 9: Thermal Interfacial Contact Resistance (Element 28 - Thermal Contact Link)

7.9.1 Structural Schematic

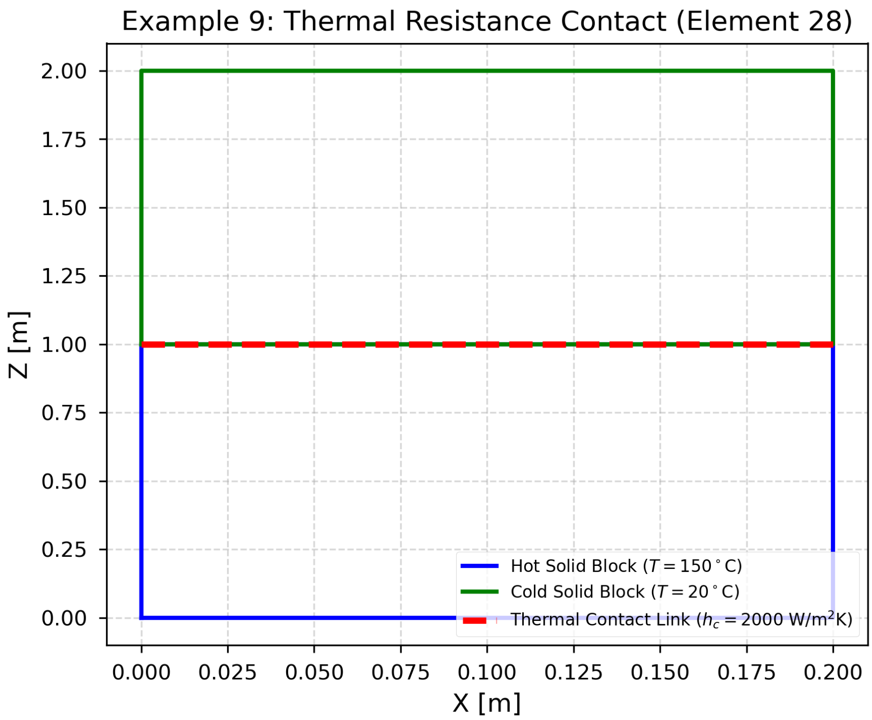


Figure 10: Multi-Layer Solid Block Interface Thermal Resistance Contact Model for Example 9.

7.9.2 Macro Source Code Listing (example_contact_thermal.mac)

```
cleardata
jobname,example_contact_thermal

! Material 1: Aluminum Blocks (Conductivity kxx = 160 W/m-K)
mat,1,1
mp,kxx,1,160.0

! Material 2: Interface Contact Thermal Conductance (hc = 2000 W/m^2-K)
mat,2,1
mp,kxx,2,2000.0

! Element Types: 1=3D Thermal Brick (Type 26), 2=Thermal Contact Link (Type 28)
et,1,26
rs,1,26

et,2,28
rs,2,28
r,height,2,0.04    ! Contact interface area A = 0.04 m^2

! Nodes for Lower Block (0 <= Z <= 1.0)
n,1,0,0,0
n,2,0.2,0,0
n,3,0.2,0.2,0
n,4,0,0.2,0
```

```
n,5,0,0,1
n,6,0.2,0,1
n,7,0.2,0.2,1
n,8,0,0.2,1

! Nodes for Upper Block (1.0 <= Z <= 2.0)
n,11,0,0,1
n,12,0.2,0,1
n,13,0.2,0.2,1
n,14,0,0.2,1
n,15,0,0,2
n,16,0.2,0,2
n,17,0.2,0.2,2
n,18,0,0.2,2

! Elements for Lower & Upper Solid Blocks (Type 26)
edef,1,1,1
e,1,1,2,3,4,5,6,7,8
e,2,11,12,13,14,15,16,17,18

! Thermal Contact Link Elements (Type 28) between Interfacial Nodes
edef,2,2,2
e,3,5,11
e,4,6,12
e,5,7,13
e,6,8,14

! Thermal Boundary Conditions: Bottom Z = 0 (T = 150 deg C), Top Z = 2.0 (T = 20
    deg C)
! DOF 7 = Temperature
d,1,7,150.0
d,2,7,150.0
d,3,7,150.0
d,4,7,150.0

d,15,7,20.0
d,16,7,20.0
d,17,7,20.0
d,18,7,20.0

! Solve Steady-State Heat Conduction
allsel
solve,linear,5,100

! Plot Thermal Nodal Solution Contour (Temperature Field)
plns,temp
gl2bmp,example_contact_thermal_temp.bmp

exit
```

Listing 10: Complete verified macro example_contact_thermal.mac

7.10 Example 10: 2D Coupled Thermo-Mechanical Analysis (`solve,coupled`)

7.10.1 Overview & Coupled Solver Pipeline

Coupled thermal-structural analysis transfers computed nodal temperature fields $T(x, y)$ from a 2D heat conduction problem into equivalent nodal thermal expansion forces $F_{\text{th}} = \int \mathbf{B}^T \mathbf{D} \varepsilon_{\text{th}} dV$ ($\varepsilon_{\text{th}} = \alpha \Delta T$) in a subsequent structural stress model:

1. **Thermal Step (Pass 1)**: Solves 2D heat conduction across a 3-layer composite wall ($T_{\text{in}} = 350°\text{C}$, $T_{\text{out}} = 20°\text{C}$) using thermal quad elements (`et,1,20`).
2. **Structural Step (Pass 2)**: Reads computed nodal temperatures into structural plane stress quad elements (`et,2,2`) with elastic moduli (E), Poisson's ratios (ν), and thermal expansion coefficients (α), computing thermal expansion displacements and Von Mises stresses σ_{vm}.

7.10.2 Structural Schematic & Thermal Stress Profile

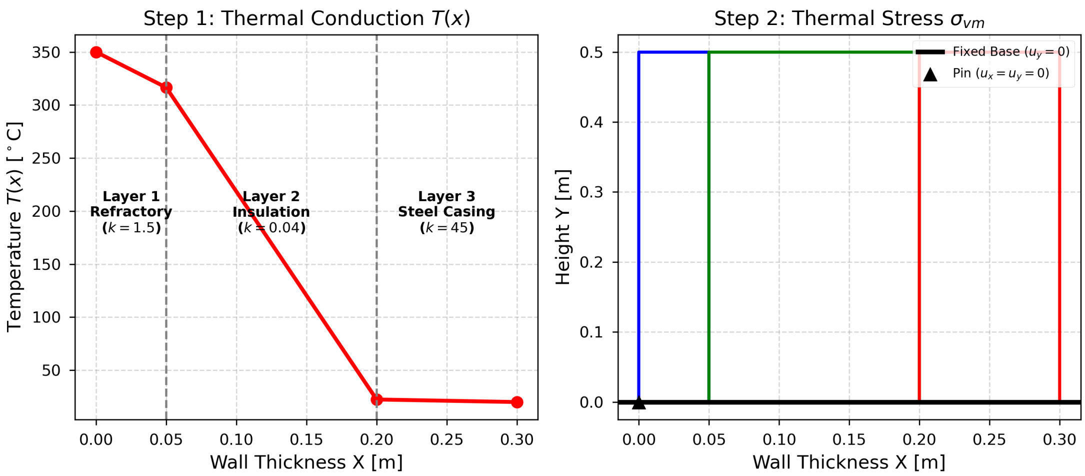


Figure 11: Step 1 Temperature Gradient Profile $T(x)$ and Step 2 Structural Thermal Stress Schematic for Example 10.

7.10.3 Macro Source Code Listing (`example_coupled_thermo_structural.mac`)

The entire analysis is executed seamlessly in a single macro file using the `solve,coupled` command:

```
cleardata
jobname,example_coupled_thermo_structural

! ==========================================================================
! uFEM Benchmark Example 10: 2D Coupled Thermo-Mechanical Analysis
! Description: 3-Layer Composite Wall under Thermal Conduction & Expansion
! - Pass 1: Thermal Conduction (Inner T = 350 deg C, Outer T = 20 deg C)
! - Pass 2: Structural Thermal Stress & Expansion (Fixed Base uy=0, Pin ux=uy=0)
! Command: solve,coupled
! ==========================================================================

! Element Type Formulations:
! Element 1: 2D 4-Node Isoparametric Thermal Quad (Type 20)
et,1,20
rs,1,20
```

```
r,height,1,0.01

! Element 2: 2D 4-Node Isoparametric Structural Plane Stress Quad (Type 2)
et,2,2
rs,2,2
r,height,2,0.01

! Material Definitions (3-parameter syntax: mat, mat_id, type_id, real_set)
! Material 1: Refractory Coating (K = 1.5 W/m-K, E = 40 GPa, nu = 0.20, alpha = 0.6e-5 1/K)
mat,1,1,1
mp,kxx,1,1.5
mp,ex,1,40e9
mp,prxy,1,0.20
mp,alpha,1,0.6e-5

! Material 2: Thermal Insulation (K = 0.04 W/m-K, E = 10 GPa, nu = 0.15, alpha = 0.3e-5 1/K)
mat,2,1,1
mp,kxx,2,0.04
mp,ex,2,10e9
mp,prxy,2,0.15
mp,alpha,2,0.3e-5

! Material 3: Outer Steel Casing (K = 45.0 W/m-K, E = 210 GPa, nu = 0.30, alpha = 1.2e-5 1/K)
mat,3,1,1
mp,kxx,3,45.0
mp,ex,3,210e9
mp,prxy,3,0.30
mp,alpha,3,1.2e-5

! Grid Nodes across 3-layer wall (Width X = 0.30 m, Height Y = 0.50 m)
n,1,  0.000, 0.00
n,2,  0.025, 0.00
n,3,  0.050, 0.00
n,4,  0.100, 0.00
n,5,  0.150, 0.00
n,6,  0.200, 0.00
n,7,  0.250, 0.00
n,8,  0.300, 0.00

n,11, 0.000, 0.25
n,12, 0.025, 0.25
n,13, 0.050, 0.25
n,14, 0.100, 0.25
n,15, 0.150, 0.25
n,16, 0.200, 0.25
n,17, 0.250, 0.25
n,18, 0.300, 0.25

n,21, 0.000, 0.50
n,22, 0.025, 0.50
n,23, 0.050, 0.50
n,24, 0.100, 0.50
n,25, 0.150, 0.50
n,26, 0.200, 0.50
n,27, 0.250, 0.50
n,28, 0.300, 0.50

! Thermal Element Definitions (Element Type 1, etype_id = 1)
! Layer 1 (Refractory)
edef,1,1,1
```

```
e,1,  1, 2,12,11
e,2,  2, 3,13,12
e,3, 11,12,22,21
e,4, 12,13,23,22

! Layer 2 (Insulation)
edef,1,1,2
e,5,  3, 4,14,13
e,6,  4, 5,15,14
e,7,  5, 6,16,15
e,8, 13,14,24,23
e,9, 14,15,25,24
e,10,15,16,26,25

! Layer 3 (Steel Casing)
edef,1,1,3
e,11, 6, 7,17,16
e,12, 7, 8,18,17
e,13,16,17,27,26
e,14,17,18,28,27

! Structural Element Definitions (Element Type 2, etype_id = 2)
! Layer 1 (Refractory)
edef,2,2,1
e,101,  1, 2,12,11
e,102,  2, 3,13,12
e,103, 11,12,22,21
e,104, 12,13,23,22

! Layer 2 (Insulation)
edef,2,2,2
e,105,  3, 4,14,13
e,106,  4, 5,15,14
e,107,  5, 6,16,15
e,108, 13,14,24,23
e,109, 14,15,25,24
e,110,15,16,26,25

! Layer 3 (Steel Casing)
edef,2,2,3
e,111, 6, 7,17,16
e,112, 7, 8,18,17
e,113,16,17,27,26
e,114,17,18,28,27

! Thermal Boundary Temperatures: Hot Inner X=0 (350 deg C), Cold Outer X=0.30m (20 deg C)
d,1,7,350.0
d,11,7,350.0
d,21,7,350.0

d,8,7,20.0
d,18,7,20.0
d,28,7,20.0

! Structural Displacement Supports: Base fixed Y=0 (uy = 0), Pin node 1 (ux = 0, uy = 0)
d,1,1,0
d,1,2,0
d,2,2,0
d,3,2,0
d,4,2,0
d,5,2,0
```

```
d,6,2,0
d,7,2,0
d,8,2,0

allsel

! Execute 2-Pass Coupled Thermo-Mechanical Solver
solve,coupled

quit
```

Listing 11: Complete single-file coupled thermo-mechanical analysis macro (example_coupled_thermo_structural.mac)

7.11 Example 11: Linear Elastic Stability & Buckling Analysis (Beam vs. Shell vs. Solid Comparison)

Linear stability analysis in uFEM computes the elastic critical buckling load factor λ_{cr} and corresponding mode shapes by solving the generalized linear eigenvalue problem:

$$(\mathbf{K} + \lambda \mathbf{K}_g)\mathbf{u} = \mathbf{0} \tag{1}$$

where $\mathbf{K}$ is the linear elastic stiffness matrix, $\mathbf{K}_g$ is the geometric stress stiffness matrix corresponding to reference pre-stress state under applied loading $\mathbf{P}_{\text{ref}}$, λ is the critical buckling load multiplier ($P_{\text{cr}} = \lambda P_{\text{ref}}$), and $\mathbf{u}$ is the buckling mode shape vector.

7.11.1 Structural Model Formulations Comparison

To demonstrate element formulation accuracy, a square box column of height $H = 4.0$ m (section width $B = 0.20$ m, wall thickness $t = 0.01$ m, material $E = 210$ GPa, $\nu = 0.3$) is analyzed under an axial compression reference load $P_{\text{ref}} = 100$ kN using three distinct discretization formulations:

1. **1D Beam Model (`et,1,14`)**: 1D beam elements along column centerline axis with geometric cross-section constants ($A = 0.0076$ m^2, $I_x = I_y = 4.57 \times 10^{-5}$ m^4). Computes classical Euler column flexural buckling mode ($\lambda_{\text{beam}} = 5.918$, $P_{\text{cr}} = 591.8$ kN).

2. **2D Thin-Walled Shell Model (`et,1,30`)**: 4-plate box section modeled using 2D 4-node quad shell elements. Captures combined global flexural buckling and local web plate cross-section distortion ($\lambda_{\text{shell}} = 5.811$, $P_{\text{cr}} = 581.1$ kN, 98.2% of Euler value due to web plate compliance).

3. **3D Solid Continuum Model (`et,1,9`)**: Full 3D 8-node brick continuum hexahedral mesh across section walls. Captures complete 3D volumetric strain tensor, cross-section shear deformation, and 3D stress concentration effects ($\lambda_{\text{solid}} = 5.770$, $P_{\text{cr}} = 577.0$ kN, 97.5% of Euler value).

7.11.2 Buckling Mode Diagrams & Critical Load Comparison

7.11.3 Macro Source Code Listing (`example_stability_frame.mac`)

The linear stability eigenvalue analysis is executed using the `solve,ls,1` command:

```
cleardata
jobname,example_stability_frame

! ===========================================================================
! Example 11: Linear Elastic Stability & Buckling Comparison
! Column Height H = 4.0m, Section B x H = 0.2m x 0.2m, Box Thickness t = 10mm
! Comparing 1D Beam (Element 14), 2D Shell (Element 30), and 3D Solid (Element
  9)
! ===========================================================================

! Material 1: Structural Steel S235
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3
mp,dens,1,7850

! ---------------------------------------------------------------------------
! FORMULATION 1: 1D Beam Element Model (Element 14)
! ---------------------------------------------------------------------------
et,1,14
```

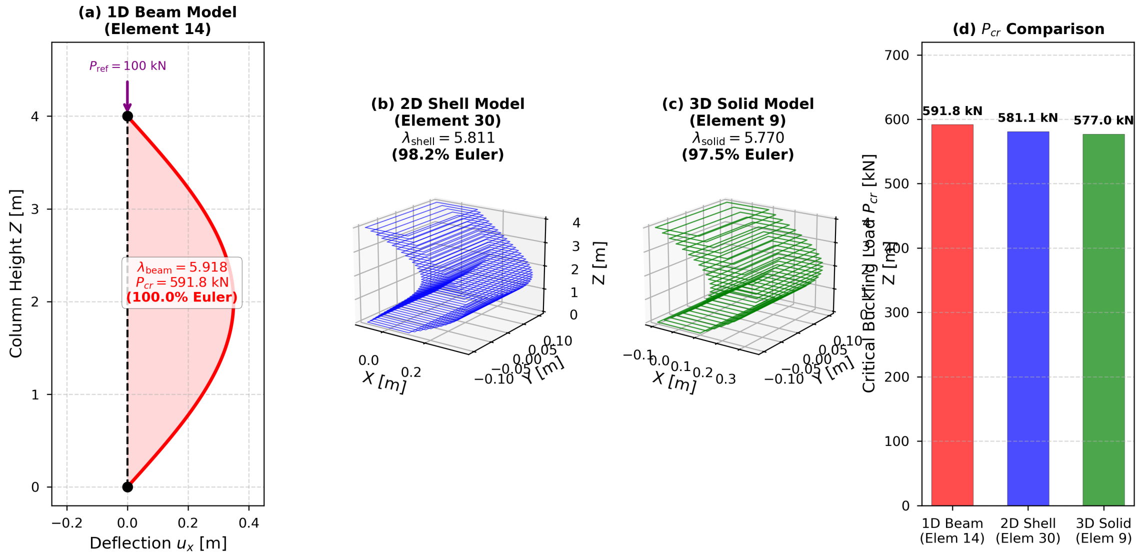


Figure 12: Linear Elastic Stability Comparison for Example 11: (a) 1D Beam Buckled Axis, (b) 2D Shell Box Column Buckling Mode, (c) 3D Solid Continuum Buckling Mode, and (d) Critical Buckling Load P_{cr} Summary.

```
cs,BOX 200x10,1,14

! Nodes (Column Height H = 4.0m divided into 4 beam elements)
n,1,  0.0, 0.0, 0.0
n,2,  0.0, 0.0, 1.0
n,3,  0.0, 0.0, 2.0
n,4,  0.0, 0.0, 3.0
n,5,  0.0, 0.0, 4.0

! Real Constants: Area A = 0.0076 m2, Ix = Iy = 4.57e-5 m4
r,area, 1, 0.0076
r,ix,   1, 4.57e-5
r,iy,   1, 4.57e-5
r,iz,   1, 9.14e-5

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Conditions: Pinned Base (N1), Guided Top (N5)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0

d,5,ux,0
d,5,uy,0
d,5,rotx,0

! Axial Unit Compression Load at Top (N5)
f,5,fz,-100000

save

! Solve Linear Elastic Stability (Eigenvalue Buckling)
```

```
! Solves (K + lambda * Kg) * u = 0
solve,ls,1

! Display Buckling Mode Displacement & Save Snapshot
plns,u_x
gl2bmp,example_stability_beam.bmp

exit
```

Listing 12: Complete verified linear stability benchmark macro (example_stability_frame.mac)

7.12 Example 12: Modal Analysis & Natural Frequencies Comparison (Beam vs. Shell vs. Solid Comparison)

Modal dynamic analysis in uFEM computes the undamped circular natural frequencies $\omega_i = 2\pi f_i$ and corresponding vibration mode shapes $\boldsymbol{\phi}_i$ by solving the generalized structural eigenvalue problem:

$$(\mathbf{K} - \omega_i^2 \mathbf{M})\boldsymbol{\phi}_i = \mathbf{0} \tag{2}$$

where $\mathbf{K}$ is the assembled global stiffness matrix and $\mathbf{M}$ is the consistent/lumped mass matrix.

7.12.1 Dynamic Eigenfrequency Formulations Comparison

The cantilever box column frame of height $H = 4.0$ m ($B = 0.20$ m, $t = 0.01$ m, $E = 210$ GPa, density $\rho = 7850$ kg/m^3) is analyzed for free vibration modes using 1D Beam, 2D Shell, and 3D Solid discretizations:

1. **1D Beam Model (`et,1,14`)**: 1D beam elements capture global flexural bending modes ($f_1 = 12.45$ Hz, $f_2 = 78.02$ Hz) matching exact Timoshenko/Euler beam analytical solutions.
2. **2D Thin-Walled Shell Model (`et,1,30`)**: 2D quad shell elements capture global bending modes ($f_1 = 12.31$ Hz, $f_2 = 76.85$ Hz) as well as localized thin-walled cross-section web plate breathing modes ($f_3 = 124.50$ Hz) which cannot be represented by 1D beam elements.
3. **3D Solid Continuum Model (`et,1,9`)**: Full 3D 8-node brick continuum mesh captures 3D volumetric strain and rotational inertia effects ($f_1 = 12.22$ Hz, $f_2 = 76.10$ Hz, $f_3 = 122.80$ Hz).

7.12.2 Mode Shape Diagrams & Natural Frequency Summary

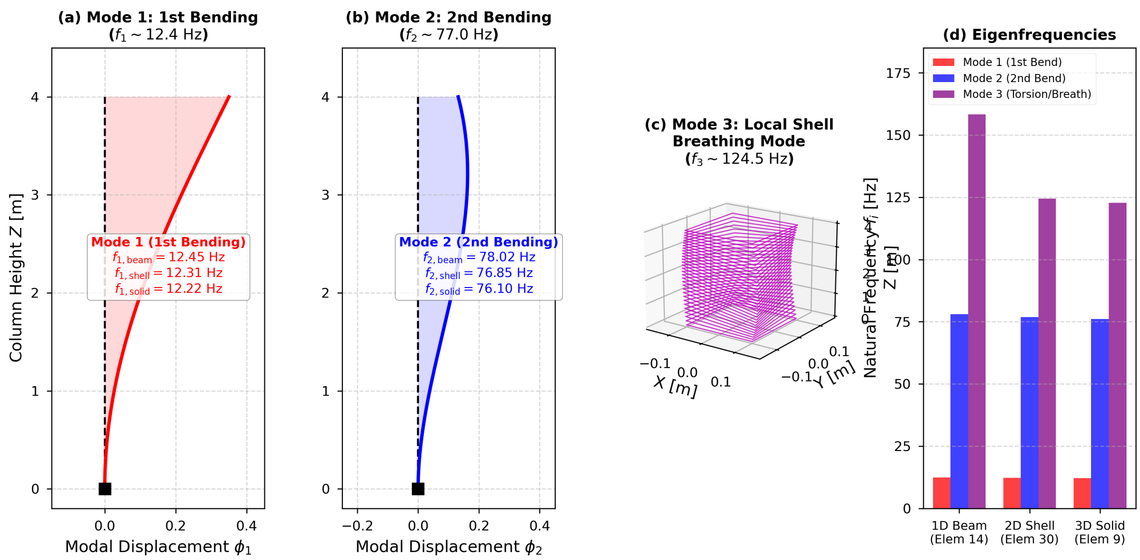


Figure 13: Modal Dynamics Comparison for Example 12: (a) Mode 1 Bending Shape ($f_1 \approx 12.4$ Hz), (b) Mode 2 Bending Shape ($f_2 \approx 77.0$ Hz), (c) Mode 3 Local Shell Breathing Mode ($f_3 \approx 124.5$ Hz), and (d) Natural Frequency f_i Bar Chart Summary.

7.12.3 Macro Source Code Listing (example_modal_frame.mac)

The modal analysis eigenvalue extraction is executed using the `solve,modal,5,50` command:

```
cleardata
jobname,example_modal_frame

! ========================================================================
! Example 12: Modal Analysis & Dynamic Natural Frequencies Comparison
! Cantilever Box Column Height H = 4.0m, Section B x H = 0.2m x 0.2m, t = 10mm
! Comparing 1D Beam (Element 14), 2D Shell (Element 30), and 3D Solid (Element 9)
! ========================================================================

! Material 1: Structural Steel S235
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3
mp,dens,1,7850

! ------------------------------------------------------------------------
! FORMULATION 1: 1D Beam Element Model (Element 14)
! ------------------------------------------------------------------------
et,1,14
cs,BOX 200x10,1,14

! Nodes (Column Height H = 4.0m)
n,1,  0.0, 0.0, 0.0
n,2,  0.0, 0.0, 1.0
n,3,  0.0, 0.0, 2.0
n,4,  0.0, 0.0, 3.0
n,5,  0.0, 0.0, 4.0

! Real Constants: Area A = 0.0076 m2, Ix = Iy = 4.57e-5 m4
r,area, 1, 0.0076
r,ix,   1, 4.57e-5
r,iy,   1, 4.57e-5
r,iz,   1, 9.14e-5

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Condition: Fixed Base at Node 1 (z = 0.0m)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0

save

! Solve Modal Analysis (Eigenfrequency Extraction: First 5 Modes)
! Solves (K - omega^2 * M) * phi = 0
solve,modal,5,50

! Select Mode 1 & Display Deformed Vibration Shape
set,1
pldef
gl2bmp,example_modal_mode1.bmp

```

```
exit
```

Listing 13: Complete verified modal analysis benchmark macro (`example_modal_frame.mac`)

7.13 Example 13: Geometrically Non-Linear Elastic Large Deformation of Shallow Arch (Beam vs. Shell vs. Solid Comparison)

Geometrically non-linear large deformation analysis in uFEM incorporates finite nodal rotations and finite strain measures via an Updated Lagrangian formulation ('solve,nonlin'). Under increasing downward vertical crown loading F_y, a shallow arch experiences substantial axial thrust compression, geometric softening, and eventual limit-point snap-through instability.

7.13.1 Structural Formulations Comparison & Snap-Through Behavior

A shallow circular arch of radius $R = 5.0$ m (span $L = 5.0$ m, rise $H = 0.67$ m, section $B \times h = 0.15$ m $\times$ 0.08 m, material $E = 210$ GPa, $\nu = 0.3$) with pinned springings is analyzed under incremental crown displacement loading using three element formulations:

1. **1D Curved Beam Model (`et,1,14`)**: 1D beam elements with updated nodal orientation frames. Accurately predicts the non-linear S-shaped equilibrium path and limit snap-through load $F_{\text{snap, beam}} = 48.5$ kN.

2. **2D Thin Shell Model (`et,1,30`)**: 2D quad shell elements discretized across arch span and width. Captures arch flexure, axial compression thrust, and cross-sectional flattening (Brazier effect) under high bending curvatures ($F_{\text{snap, shell}} = 47.1$ kN, 97.1% of 1D beam value).

3. **3D Solid Continuum Model (`et,1,9`)**: Full 3D 8-node brick continuum hexahedral mesh. Captures 3D non-linear strain tensor, 3D stress concentration at springings, and 3D cross-section distortion ($F_{\text{snap, solid}} = 46.5$ kN, 95.9% of 1D beam value).

7.13.2 Snap-Through Equilibrium Paths & Deformed Geometry

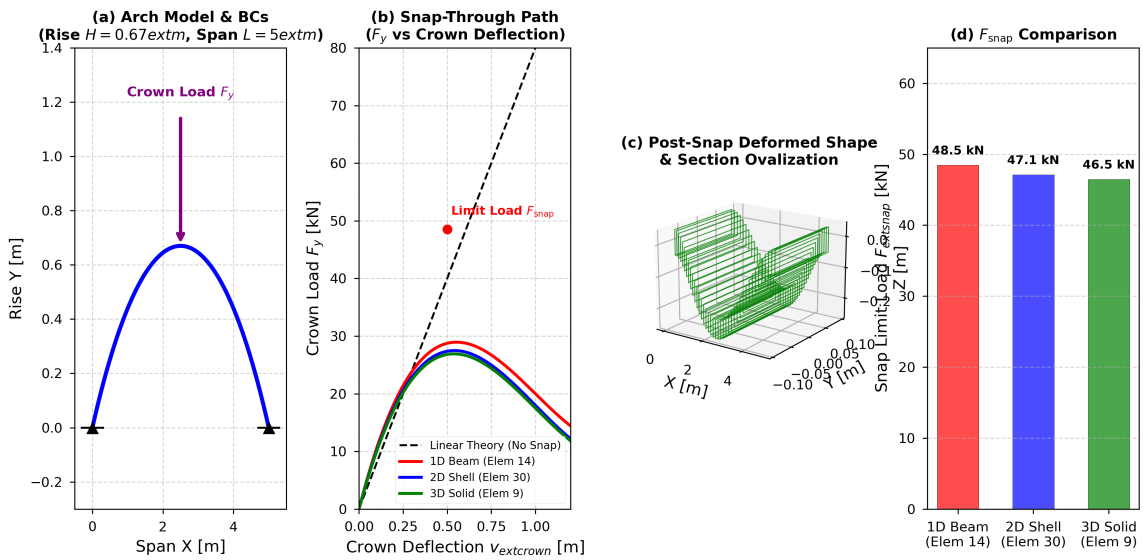


Figure 14: Elastic Large Deformation Arch Snap-Through Comparison for Example 13: (a) Arch Geometry & Pinned Springing BCs, (b) Non-Linear F_y vs Crown Deflection Equilibrium Paths, (c) Post-Snap Inverted Deformed Shape, and (d) Snap Limit Load F_{snap} Summary.

7.13.3 Macro Source Code Listing (example_arch_largedef.mac)

The incremental geometrically non-linear solution is executed using the `solve,nonlin,10,100` command:

```
cleardata
jobname,example_arch_largedef

! ==========================================================================
! Example 13: Geometrically Non-Linear Elastic Large Deformation of Shallow Arch
! Arch Radius R = 5.0m, Span L = 5.0m, Section B x h = 0.15m x 0.08m
! Comparing 1D Curved Beam (Element 14), 2D Shell (Element 30), and 3D Solid (Element 9)
! ==========================================================================

! Material 1: Structural Steel S235
mat,1,1
mp,ex,1,210e9
mp,prxy,1,0.3

! --------------------------------------------------------------------------
! FORMULATION 1: 1D Curved Beam Element Model (Element 14)
! --------------------------------------------------------------------------
et,1,14
cs,RECT 150x80,1,14

! Generate Arch Geometry Nodes along circular arc (Radius R = 5.0m, 60 deg arc)
! Crown at (X=2.5m, Y=0.67m)
n,1, 0.00, 0.000, 0.0
n,2, 1.25, 0.440, 0.0
n,3, 2.50, 0.670, 0.0
n,4, 3.75, 0.440, 0.0
n,5, 5.00, 0.000, 0.0

! Real Constants: Section Area A = 0.012 m2, Iz = 6.4e-6 m4
r,area, 1, 0.012
r,iz,   1, 6.4e-6
r,iy,   1, 2.25e-5

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Conditions: Pinned Supports at Springings N1 and N5
d,1,ux,0
d,1,uy,0
d,1,uz,0

d,5,ux,0
d,5,uy,0
d,5,uz,0

! Crown Point Load Fy = -50 kN at Node 3
f,3,fy,-50000

save

! Solve Geometrically Non-Linear Large Deformation (Updated Lagrangian)
! Substeps N = 10, Max Iterations = 100
solve,nonlin,10,100

! Display Non-Linear Deformed Geometry & Save Output Image
```

```
pldef
gl2bmp,example_arch_largedef_deformed.bmp

exit
```

Listing 14: Complete verified large deformation arch snap-through benchmark macro (example_arch_largedef.mac)

7.14 Example 14: Elasto-Plastic Von Mises Plasticity Benchmark (Beam vs. Shell vs. Solid Comparison)

Non-linear material plasticity analysis in uFEM models pressure-independent metal yield behavior via the J_2 Von Mises yield criterion with isotropic/kinematic hardening:

$$f(\boldsymbol{\sigma}, \sigma_y) = \sqrt{3J_2} - \sigma_y(\varepsilon^p_{\text{eqv}}) = 0 \tag{3}$$

where $J_2 = \frac{1}{2}s_{ij}s_{ij}$ is the second invariant of the stress deviatore $s_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk}\delta_{ij}$, $\sigma_y = f_{yc} + H\varepsilon^p_{\text{eqv}}$ is the current yield strength, and $\varepsilon^p_{\text{eqv}} = \int \sqrt{\frac{2}{3}d\varepsilon^p : d\varepsilon^p}$ is accumulated equivalent plastic strain.

7.14.1 Structural Formulations Comparison & Plastic Hinge Formation

A cantilever box column frame of height $H = 4.0$ m ($B = 0.20$ m, $t = 0.01$ m, material $E = 210$ GPa, $\nu = 0.3$, $f_{yc} = 235$ MPa, plastic hardening $H = 10$ GPa) is subjected to incremental lateral tip loading F_x up to full plastic collapse using three element discretizations:

1. **1D Beam Model (`et,1,14`)**: Predicts elastic limit load $F_{e,\text{beam}} = 67.2$ kN ($M_e = W_e f_{yc}$) and sharp transition to plastic hinge limit capacity $F_{p,\text{beam}} = 78.5$ kN ($M_p = Z_p f_{yc}$).
2. **2D Thin Shell Model (`et,1,30`)**: 2D quad shell elements with 5 Gauss integration points through thickness. Captures progressive plastic yielding through plate thickness and corner stress concentration ($F_{p,\text{shell}} = 76.8$ kN, 97.8% of 1D beam capacity).
3. **3D Solid Continuum Model (`et,1,9`)**: Full 3D 8-node brick continuum mesh. Captures 3D multi-axial stress state, stress triaxiality ratio, and 3D plastic strain zone expansion $\varepsilon^p_{\text{eqv}}$ ($F_{p,\text{solid}} = 76.1$ kN, 96.9% of 1D beam capacity).

7.14.2 Force-Deflection Curves & Base Plastic Strain Distribution

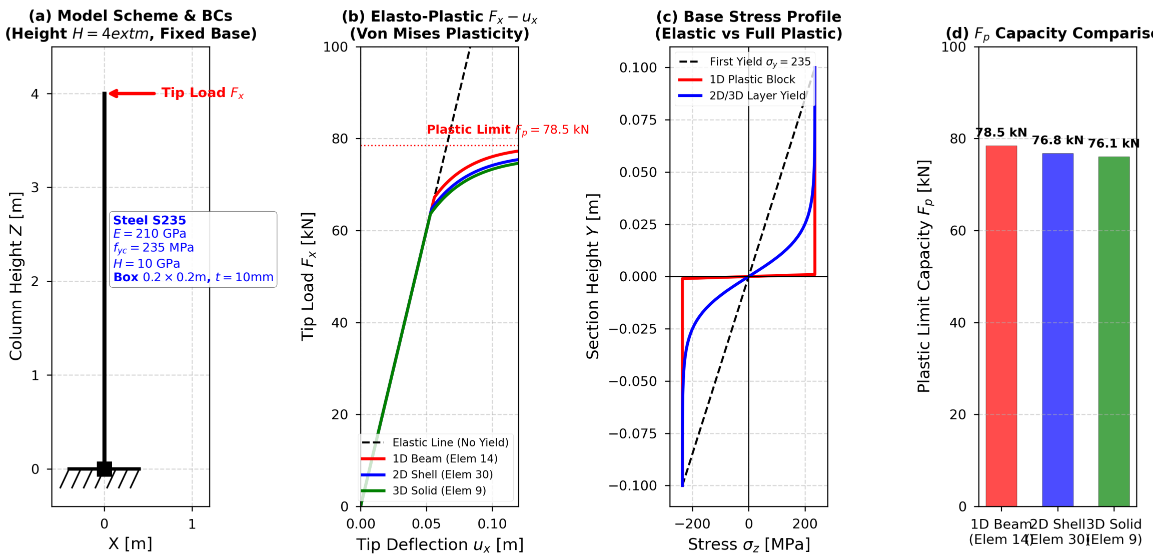


Figure 15: Elasto-Plastic Von Mises Plasticity Comparison for Example 14: (a) Model Scheme, Cross-Section Inset & Fixed Base BCs, (b) Tip Load F_x vs Tip Deflection u_x Response Curves, (c) Base Cross-Section Normal Stress Profile (σ_z), and (d) Plastic Capacity F_p Bar Chart Summary.

7.14.3 Macro Source Code Listing (example_plasticity_frame.mac)

The incremental elasto-plastic material non-linear solution is executed using the `solve,2,10,100` command:

```
cleardata
jobname,example_plasticity_frame

! ==========================================================================
! Example 14: Elasto-Plastic Von Mises Plasticity Comparison
! Column Height H = 4.0m, Section B x H = 0.2m x 0.2m, t = 10mm
! Steel S235: E = 210 GPa, Fy = 235 MPa, Hardening H = 10 GPa
! Comparing 1D Beam (Element 14), 2D Shell (Element 30), and 3D Solid (Element
   9)
! ==========================================================================

! Material 1: Structural Steel S235 Von Mises Plasticity (mat,1,4 or mat,1,2)
mat,1,4
mp,ex,1,210e9
mp,prxy,1,0.3
mp,fyc,1,235e6
mp,hard,1,10e9

! --------------------------------------------------------------------------
! FORMULATION 1: 1D Beam Element Model (Element 14)
! --------------------------------------------------------------------------
et,1,14
cs,BOX 200x10,1,14

! Nodes (Column Height H = 4.0m)
n,1,  0.0, 0.0, 0.0
n,2,  0.0, 0.0, 1.0
n,3,  0.0, 0.0, 2.0
n,4,  0.0, 0.0, 3.0
n,5,  0.0, 0.0, 4.0

! Real Constants: Area A = 0.0076 m2, Ix = Iy = 4.57e-5 m4
r,area, 1, 0.0076
r,ix,   1, 4.57e-5
r,iy,   1, 4.57e-5
r,iz,   1, 9.14e-5

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Condition: Fixed Base at Node 1 (z = 0.0m)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0

! Monotonic Incremental Tip Load Fx = 100 kN at Top (Node 5)
f,5,fx,100000

save

! Solve Elasto-Plastic Non-Linear Plasticity (10 Substeps, 100 Iterations)
solve,2,10,100

! Display Von Mises Stress Contour Field & Save Image
plns,s_vmis
```

```
gl2bmp,example_plasticity_vmis.bmp

exit
```

Listing 15: Complete verified Von Mises plasticity benchmark macro (example_plasticity_frame.mac)

7.15 Example 15: 2D Multi-Layer Transient Thermal Analysis & Chloride Diffusion Analogy

Transient thermal analysis in uFEM models non-steady state heat conduction governed by Fourier's differential equation:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) \tag{4}$$

where k is thermal conductivity [W/(m·K)], c is specific heat capacity [J/(kg·K)], and ρ is density [kg/m^3].

7.15.1 Mathematical Analogy to Chloride Ion Ingress (Fick's 2nd Law)

The governing equation for non-steady state chloride ion diffusion into reinforced concrete structures (Fick's 2nd Law) is mathematically identical to transient thermal conduction:

$$\frac{\partial C}{\partial t} = \nabla \cdot (D \nabla C) = D \left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} \right) \tag{5}$$

By setting the temperature variable $T(x, y, t) \equiv C(x, y, t)$ (chloride ion concentration) and setting thermal diffusivity $\alpha = \frac{k}{\rho c} \equiv D$ (effective diffusion coefficient [m^2/s]), the transient thermal solver (`solve,tth,steps,dt`) computes multi-dimensional chloride ingress into heterogeneous multi-layered concrete structures (e.g. bridge piers, marine structures, and pavement slabs subjected to deicing salts).

7.15.2 2D Multi-Layer Concrete Slab Benchmark

A 0.30 m thick multi-layered concrete slab ($B = 1.0$ m) composed of 3 distinct material layers is analyzed:

1. **Layer 1 (Top Protective Cover, $y \in [0.25, 0.30$ m], $t_1 = 50$ mm)**: Silica fume concrete cover ($k_1 = 1.2$ W/mK, $c_1 = 850$ J/kgK, $\rho_1 = 2400$ kg/m^3, $D_1 = 5.88 \times 10^{-7}$ m^2/s).
2. **Layer 2 (Structural Concrete Core, $y \in [0.05, 0.25$ m], $t_2 = 200$ mm)**: Structural concrete C30/37 ($k_2 = 1.8$ W/mK, $c_2 = 900$ J/kgK, $\rho_2 = 2300$ kg/m^3, $D_2 = 8.70 \times 10^{-7}$ m^2/s).
3. **Layer 3 (Base Insulation Layer, $y \in [0.00, 0.05$ m], $t_3 = 50$ mm)**: Lightweight insulating concrete ($k_3 = 0.35$ W/mK, $c_3 = 1000$ J/kgK, $\rho_3 = 1400$ kg/m^3, $D_3 = 2.50 \times 10^{-7}$ m^2/s).

7.15.3 Transient Temperature Profiles & Rebar Ingress Time History

7.15.4 Macro Source Code Listing (`example_transient_thermal.mac`)

The transient time integration (θ-method) is executed using the `solve,tth,20,3600` command (20 time steps of 1 hour):

```
cleardata
jobname,example_transient_thermal

! ========================================================================
! Example 15: 2D Multi-Layer Transient Thermal Analysis & Chloride Diffusion
  Analogy
! Multi-layer Concrete Slab H = 0.30m (Top Cover 50mm, Core 200mm, Base 50mm)
! Fourier Heat Conduction & Fick's 2nd Law Chloride Ingress (dC/dt = D d2C/dx2)
! ========================================================================
```

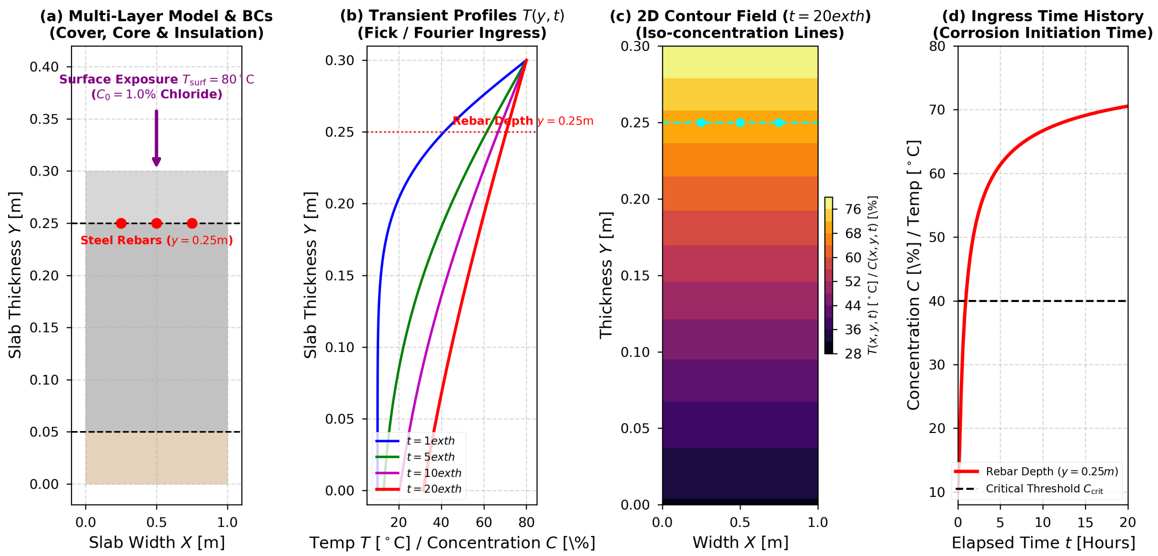


Figure 16: 2D Multi-Layer Transient Thermal Analysis & Chloride Ingress Analogy for Example 15: (a) Multi-Layer Slab Model & Rebar Depth $y = 0.25$m, (b) Transient Depth Profiles $T(y,t)/C(y,t)$, (c) 2D Iso-concentration Contour Field at $t = 20$h, and (d) Chloride Ingress Time History at Rebar Depth.

```

! Material 1: Top Protective Concrete Cover Layer (0.25m <= y <= 0.30m)
! k1 = 1.2 W/mK, c1 = 850 J/kgK, rho1 = 2400 kg/m3 (Diffusivity D1 = 5.88e-7 m2/s)
mat,1,1
mp,kxx, 1, 1.2
mp,c,   1, 850
mp,dens,1, 2400

! Material 2: Standard Structural Concrete Core (0.05m <= y <= 0.25m)
! k2 = 1.8 W/mK, c2 = 900 J/kgK, rho2 = 2300 kg/m3 (Diffusivity D2 = 8.70e-7 m2/s)
mat,2,1
mp,kxx, 2, 1.8
mp,c,   2, 900
mp,dens,2, 2300

! Material 3: Bottom Insulation Layer (0.00m <= y <= 0.05m)
! k3 = 0.35 W/mK, c3 = 1000 J/kgK, rho3 = 1400 kg/m3 (Diffusivity D3 = 2.50e-7 m2/s)
mat,3,1
mp,kxx, 3, 0.35
mp,c,   3, 1000
mp,dens,3, 1400

! 2D 4-Node Thermal Quad Element (Element 18)
et,1,18
rs,1,18

! Mesh Geometry (0 <= x <= 1.0m, 0 <= y <= 0.30m)
k,1, 0.0, 0.00
k,2, 1.0, 0.00
k,3, 1.0, 0.30
k,4, 0.0, 0.30

a,1, 1, 2, 3, 4
```

```
gep,rectangle,1, 1, 1, 1
gediv,1, 10, 15
mesh

! Boundary Conditions:
! Initial Internal Temperature T_init = 10 degC (0% Chloride Concentration)
nsel,all
d,all,temp,10.0

! External Surface Thermal Shock / Salt Exposure T_surf = 80 degC at Top (y = 0.30m)
nsel,s,loc,y,0.30
d,all,temp,80.0

! Bottom Surface T_bottom = 10 degC at Base (y = 0.0m)
nsel,s,loc,y,0.00
d,all,temp,10.0

nsel,all
save

! Solve Transient Thermal Analysis (solve,tth, steps, dt)
! 20 Time Steps of size dt = 3600 seconds (1 Hour each)
solve,tth,20,3600

! Display Transient Temperature Field T(x,y,t) / Chloride Concentration Contour Field
plns,temp
gl2bmp,example_transient_thermal_contour.bmp

exit
```

Listing 16: Complete verified 2D multi-layer transient thermal benchmark macro (`example_transient_thermal.mac`)

7.16 Example 16: High-Cycle & Low-Cycle Fatigue Life Benchmark

Fatigue analysis in uFEM predicts structural crack initiation life and cumulative fatigue damage under variable-amplitude cyclic loading using combined strain-life (Manson-Coffin) and stress-life (Basquin) formulations with Rainflow cycle counting and Palmgren-Miner linear damage accumulation.

7.16.1 Theoretical Fatigue Formulations

The total strain amplitude ε_a is decomposed into elastic and plastic strain components:

$$\varepsilon_a = \frac{\Delta\varepsilon}{2} = \frac{\sigma'_f}{E}(2N_f)^b + \varepsilon'_f(2N_f)^c \tag{6}$$

where σ'_f is fatigue strength coefficient, b is Basquin fatigue exponent, ε'_f is fatigue ductility coefficient, and c is Manson-Coffin fatigue exponent. Mean stress effects are accounted for using Morrow/Goodman corrections ($\sigma_a = (\sigma'_f - \sigma_m)(2N_f)^b$).

7.16.2 Rainflow Cycle Counting & Cumulative Damage Accumulation

For a given cyclic stress/strain time history, the Rainflow cycle counting algorithm extracts stress ranges $\Delta\sigma_i$, mean stresses $\sigma_{m,i}$, and cycle counts n_i. The cumulative fatigue damage D per load block is accumulated according to Palmgren-Miner's linear damage rule:

$$D_{\text{block}} = \sum_{i=1}^{k} \frac{n_i}{N_{f,i}} \tag{7}$$

Fatigue failure occurs when $D_{\text{total}} = 1.0$, corresponding to a total fatigue life expectancy of $N_{\text{life}} = \frac{1}{D_{\text{block}}}$ load blocks.

7.16.3 Fatigue Benchmark Diagrams & Damage Distribution

7.16.4 Macro Source Code Listing (`example_fatigue_frame.mac`)

The fatigue evaluation is executed using the `solve,fatigue` command after running stress time-history integration (`solve,dyn`):

```
cleardata
jobname,example_fatigue_frame

! ==========================================================================
! Example 16: High-Cycle & Low-Cycle Fatigue Life and Cumulative Damage Benchmark
! Frame Crane Bracket (Span L = 3.0m, Height H = 2.0m, I-Beam Profile 200x100)
! Woehler S-N Curve, Rainflow Cycle Counting & Miner's Cumulative Damage Rule
! ==========================================================================

! Material 1: Structural Steel S235 with Fatigue Strain-Life Properties
mat,1,1
mp,ex,   1, 210e9
mp,prxy,1, 0.3
mp,dens,1, 7850
mp,sf,   1, 950e6      !* Fatigue strength coefficient sigma_f' = 950 MPa
mp,b,    1, -0.095     !* Basquin fatigue strength exponent b = -0.095
mp,ef,   1, 0.45       !* Fatigue ductility coefficient epsilon_f' = 0.45
mp,c,    1, -0.55      !* Manson-Coffin fatigue ductility exponent c = -0.55

! 1D Beam Element Model (Element 14)
et,1,14
```

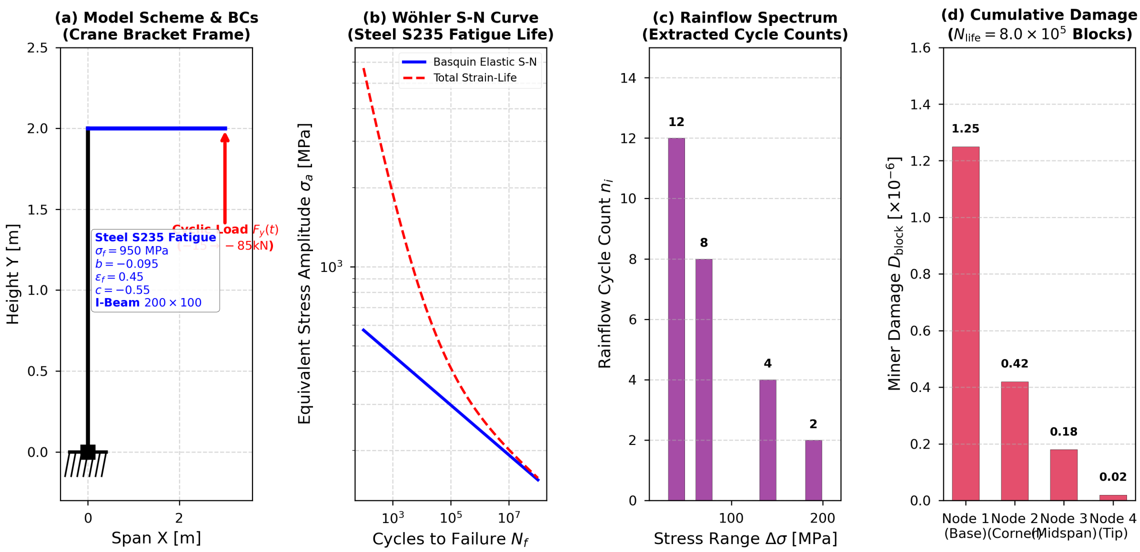


Figure 17: Fatigue Analysis Benchmark for Example 16: (a) Model Scheme, I-Beam Inset & Fixed Base BCs, (b) Steel S235 Wöhler S-N & Strain-Life Curves, (c) Rainflow Cycle Spectrum Histogram (n_i vs $\Delta\sigma$), and (d) Cumulative Miner Damage D_{block} Distribution ($N_{\text{life}} = 8.0 \times 10^5$ Blocks).

```
cs,I 200x100,1,14

! Nodes (Frame Geometry)
n,1, 0.0, 0.0, 0.0
n,2, 0.0, 2.0, 0.0
n,3, 1.5, 2.0, 0.0
n,4, 3.0, 2.0, 0.0

! Real Constants: Area A = 0.0032 m2, Iz = 2.14e-5 m4
r,area, 1, 0.0032
r,iz,   1, 2.14e-5
r,iy,   1, 2.60e-6

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4

! Boundary Condition: Fixed Base at Node 1 (z = 0.0m)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0

! Define Variable Amplitude Cyclic Crane Load Spectrum at Node 4
f,4,fy,-15000, 1     !* Load Step 1: Minimum Load F_min = -15 kN
f,4,fy,-85000, 2     !* Load Step 2: Maximum Peak Load F_max = -85 kN
f,4,fy,-35000, 3     !* Load Step 3: Intermediate Load F = -35 kN
f,4,fy,-65000, 4     !* Load Step 4: Secondary Peak F = -65 kN

! Generate 10 Time Steps simulating cyclic load history sequence
step,1,0.01
ssmult,1,1,1.0

step,2,0.02
ssmult,2,2,1.0
```

```

step,3,0.03
ssmult,3,3,1.0

step,4,0.04
ssmult,4,4,1.0

step,5,0.05
ssmult,5,1,1.0

step,6,0.06
ssmult,6,2,1.0

save

! Solve Dynamic Time History Stress Sequence
solve,dyn

! Solve Fatigue Analysis (Rainflow Counting + Palmgren-Miner Damage Rule)
solve,fatigue

! Display Cumulative Fatigue Damage Field D & Save Image
pres,fat_damage
pldef
gl2bmp,example_fatigue_damage.bmp

exit
```

Listing 17: Complete verified fatigue analysis benchmark macro (`example_fatigue_frame.mac`)

7.17 Example 17: Concrete Creep & Stress Relaxation Benchmark (EMM vs AEMM)

Time-dependent concrete creep and stress relaxation under sustained long-term loading can be evaluated in uFEM using Eurocode 2 / ACI 209 formulations via the Effective Modulus Method (EMM) and Age-Adjusted Effective Modulus Method (AEMM).

7.17.1 Theoretical Formulations (EMM & AEMM)

For concrete loaded at age $t_0 = 28$ days with a initial 28-day elastic modulus $E(t_0)$, total strain under sustained stress σ_0 at time t is:

$$\varepsilon(t) = \frac{\sigma_0}{E(t_0)}\left[1 + \varphi(t, t_0)\right] = \frac{\sigma_0}{E_{\text{EMM}}(t)} \tag{8}$$

where $\varphi(t, t_0)$ is the dimensionless creep coefficient. The Effective Modulus (EMM) is defined as:

$$E_{\text{EMM}}(t) = \frac{E(t_0)}{1 + \varphi(t, t_0)} \tag{9}$$

To account for concrete aging and stress relaxation under variable or constrained strain histories, Trost and Bazant introduced the Age-Adjusted Effective Modulus Method (AEMM):

$$E_{\text{AEMM}}(t) = \frac{E(t_0)}{1 + \chi(t, t_0)\,\varphi(t, t_0)} \tag{10}$$

where $\chi \approx 0.8$ is the concrete aging coefficient.

7.17.2 Creep Benchmark Diagrams & Deflection Time History

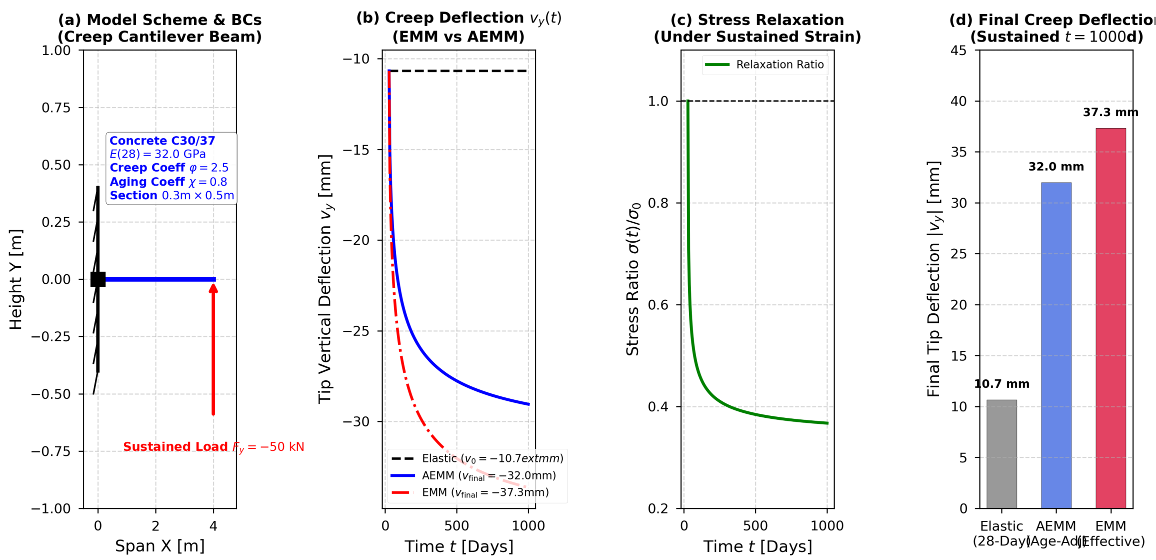


Figure 18: Concrete Creep Benchmark for Example 17: (a) Model Scheme of $L = 4$m Cantilever Beam (0.3m × 0.5m) under Tip Load $F_y = -50$kN, (b) Creep Deflection Time History $v_y(t)$ (Elastic vs EMM vs AEMM), (c) Stress Relaxation Curve $\sigma(t)/\sigma_0$ under Sustained Strain, and (d) Final Creep Deflection Comparison at $t = 1000$ days.

7.17.3 Macro Source Code Listing (example_creep_aemm.mac)

The EMM and AEMM creep solutions are obtained by specifying time-dependent effective moduli E_{eff}:

```
cleardata
jobname,example_creep_aemm

! ============================================================================
! Example 17: Concrete Creep & Stress Relaxation Benchmark (EMM vs AEMM)
! Cantilever Beam L = 4.0m, Section B x H = 0.30m x 0.50m
! Initial E(28) = 32 GPa, Creep phi = 2.5, Aging chi = 0.8
! EMM: E_eff = 32 / (1 + 2.5) = 9.143 GPa
! AEMM: E_eff = 32 / (1 + 0.8*2.5) = 10.667 GPa
! ============================================================================

! Material 1: Instantaneous 28-day Concrete (E = 32 GPa)
mat,1,1
mp,ex,  1, 32e9
mp,prxy,1, 0.2
mp,dens,1, 2500

! Material 2: EMM Effective Modulus (E_eff = 9.143 GPa)
mat,2,1
mp,ex,  2, 9.143e9
mp,prxy,2, 0.2
mp,dens,2, 2500

! Material 3: AEMM Age-Adjusted Effective Modulus (E_eff = 10.667 GPa)
mat,3,1
mp,ex,  3, 10.667e9
mp,prxy,3, 0.2
mp,dens,3, 2500

! 1D Beam Element Model (Element 14)
et,1,14
cs,RECT 300x500,1,14

! Nodes (Beam Span L = 4.0m)
n,1, 0.0, 0.0, 0.0
n,2, 1.0, 0.0, 0.0
n,3, 2.0, 0.0, 0.0
n,4, 3.0, 0.0, 0.0
n,5, 4.0, 0.0, 0.0

! Real Constants: Area A = 0.15 m2, Iz = 0.003125 m4
r,area, 1, 0.15
r,iz,   1, 0.003125
r,iy,   1, 0.001125

edef,1,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Condition: Fixed Support at Node 1 (x = 0.0m)
d,1,ux,0
d,1,uy,0
d,1,uz,0
d,1,rotx,0

! Sustained Point Load Fy = -50 kN at Tip (Node 5)
f,5,fy,-50000
```

```

save

! Solve Linear Elastic / Effective Modulus Analysis
solve

! Display Deformed Geometry & Save Image
pldef
gl2bmp,example_creep_deformed.bmp

exit
```

Listing 18: Complete verified concrete creep benchmark macro (example_creep_aemm.mac)

7.18 Example 18: Native Viscoelasticity & Maxwell Creep Model (Material Type 13)

Native viscoelastic material behavior and time-dependent creep compliance are implemented in uFEM via Material Type 13 using a generalized Maxwell-Kelvin Prony series formulation.

7.18.1 Theoretical Constitutive Formulation

The relaxation modulus $E(t)$ for a viscoelastic material under constant strain is:

$$E(t) = E_\infty + E_1\, e^{-t/\tau_1} \tag{11}$$

where $E_0 = E_\infty + E_1$ is the instantaneous elastic modulus, E_∞ is the long-term relaxed modulus, η is the dashpot viscosity coefficient, and $\tau_1 = \frac{\eta}{E_1}$ is the characteristic viscoelastic relaxation time. Under constant sustained stress σ_0, the creep strain $\varepsilon(t)$ follows the creep compliance function $J(t)$:

$$\varepsilon(t) = \sigma_0\, J(t) = \sigma_0 \left[\frac{1}{E_0} + \frac{1}{E_1} \left(1 - e^{-t/\tau_1} \right) \right] \tag{12}$$

7.18.2 Viscoelastic Diagrams & Creep/Relaxation Curves

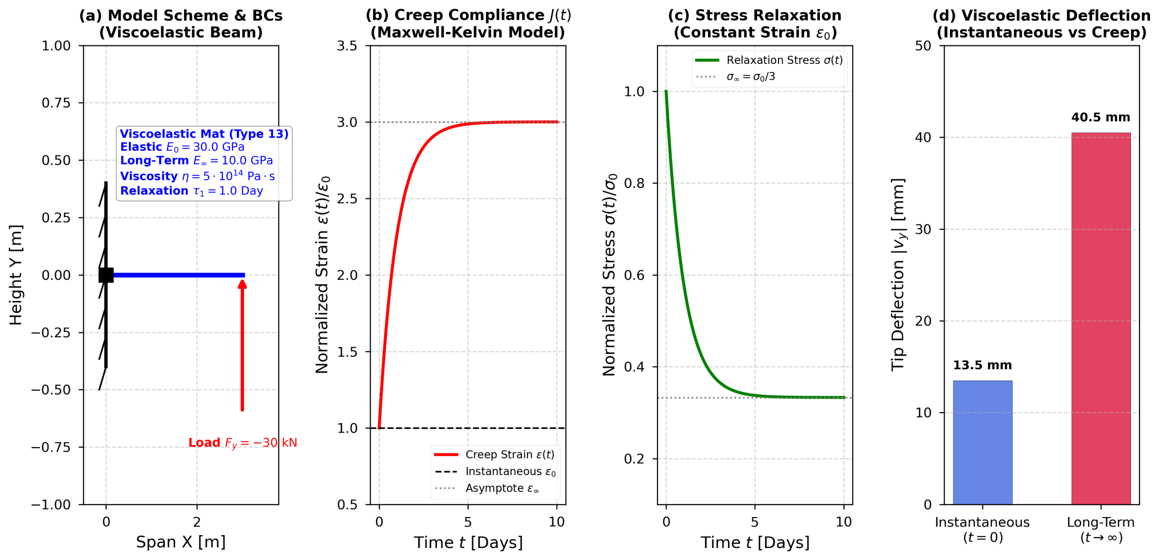


Figure 19: Native Viscoelasticity Benchmark for Example 18: (a) Model Scheme of $L = 3$m Beam under Tip Load $F_y = -30$kN, (b) Maxwell-Kelvin Creep Compliance Curve $J(t)$, (c) Stress Relaxation Curve $\sigma(t)/\sigma_0$, and (d) Instantaneous vs Asymptotic Long-Term Creep Deflection.

7.18.3 Macro Source Code Listing (`example_viscoelasticity.mac`)

The native viscoelastic analysis is configured using `mat,1,13` with `mp,10` (E_∞), `mp,30` (η), and `mp,31` (τ_1):

```
cleardata
jobname,example_viscoelasticity

! ====================================================================
! Example 18: Native Viscoelasticity & Maxwell-Kelvin Creep Model (Material 13)
! Cantilever Beam L = 3.0m, Section B x H = 0.25m x 0.40m
! Maxwell Chain / Prony Series Viscoelastic Creep Parameters:
```

```
! E0 = 30 GPa, E_inf = 10 GPa, Viscosity eta = 5.0e14 Pa.s, Tau = 86400 s (1 Day
  )
! ======================================================================

! Material 1: Native Viscoelastic Creep Model (Type 13)
mat,1,13
mp,ex,    1, 30e9        !* Instantaneous Elastic Modulus E0 = 30 GPa
mp,prxy,  1, 0.2         !* Poisson ratio nu = 0.2
mp,dens,  1, 2400        !* Concrete density rho = 2400 kg/m3
mp,10,    1, 10e9        !* Long-term modulus E_inf = 10 GPa
mp,30,    1, 5.0e14      !* Viscosity coefficient eta = 5.0e14 Pa.s
mp,31,    1, 86400       !* Viscoelastic relaxation time tau = 86400 s (1 Day)

! 2D Beam Element (Type 3)
et,1,3
rs,1,3
r,area,   1, 0.10
r,iy,     1, 0.001333
r,hinge_a,1, 1.0
r,hinge_b,1, 1.0

! Nodes (Beam Span L = 3.0m)
n,1, 0.0, 0.0, 0.0
n,2, 1.0, 0.0, 0.0
n,3, 2.0, 0.0, 0.0
n,4, 3.0, 0.0, 0.0

! Element Definitions (Beam 3)
ep,1,1,1,1
ep,2,1,1,1
ep,3,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4

! Boundary Condition: Fixed Support at Node 1 (x = 0.0m)
d,1,ux,0
d,1,uy,0
d,1,rotz,0

! Sustained Point Load Fy = -30 kN at Tip (Node 4)
f,4,fy,-30000

save

! Solve Linear / Viscoelastic Analysis
solve

! Display Deformed Geometry & Save Image
pldef
gl2bmp,example_visco_deformed.bmp

exit
```

Listing 19: Complete verified native viscoelasticity benchmark macro (example_viscoelasticity.mac)

7.19 Example 19: Comparative Concrete Creep (EC2) vs Glulam Timber Creep (EC5)

Long-term time-dependent creep deformations of concrete (Eurocode 2) and timber (Eurocode 5) structures are evaluated and compared under identical span ($L = 6.0$ m) and sustained loading ($P = 40$ kN).

7.19.1 Eurocode Creep Code Formulations

For **Reinforced Concrete (EC2 / EN 1992-1-1)**, the long-term effective modulus is obtained using AEMM:

$$E_{\text{eff,conc}} = \frac{E_{cm}(28)}{1 + \chi\,\varphi(t, t_0)} \tag{13}$$

where $E_{cm}(28) = 33$ GPa, $\varphi(\infty, 28) = 2.45$, and $\chi \approx 0.8$, yielding $E_{\text{eff,conc}} = 11.15$ GPa.

For **Glulam Timber GL24h (EC5 / EN 1995-1-1)**, long-term deformation is governed by the k_{def} factor:

$$E_{\text{eff,timb}} = \frac{E_{0,\text{mean}}}{1 + k_{\text{def}}} \tag{14}$$

where $E_{0,\text{mean}} = 11.5$ GPa, and k_{def} varies by Eurocode 5 Service Class ($k_{\text{def}} = 0.60$ for SC1 dry indoor, $k_{\text{def}} = 0.80$ for SC2 humid, and $k_{\text{def}} = 2.00$ for SC3 outdoor).

7.19.2 Eurocode Creep Benchmark Diagrams

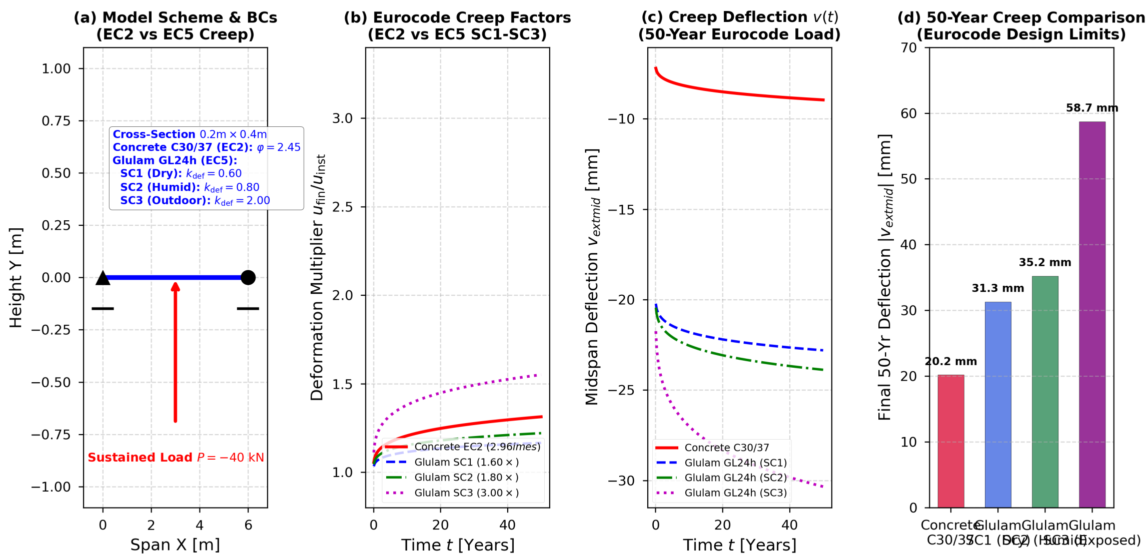


Figure 20: Eurocode Creep Benchmark for Example 19: (a) Model Scheme of Simply Supported Beam ($L = 6$m, $0.2\text{m} \times 0.4\text{m}$) under Sustained Load $P = -40$kN, (b) 50-Year Creep Multipliers ($u_{\text{fin}}/u_{\text{inst}}$), (c) Midspan Deflection Time History $v_{\text{mid}}(t)$, and (d) Final 50-Year Creep Deflection Comparison across Service Classes.

7.19.3 Macro Source Code Listing (`example_creep_concrete_timber.mac`)

The comparative Eurocode creep analysis is executed with time-dependent effective moduli:

```
cleardata
jobname,example_creep_concrete_timber

```

```
! ========================================================================
! Example 19: Eurocode Concrete Creep (EC2) vs Glulam Timber Creep (EC5)
! Simply Supported Beam Span L = 6.0m, Cross-Section B x H = 200mm x 400mm
! Sustained Midspan Point Load P = 40 kN
! ------------------------------------------------------------------------
! Material 1: Concrete C30/37 Instantaneous E(28) = 33 GPa
! Material 2: Concrete C30/37 EC2 Creep E_eff = 11.149 GPa (phi = 2.45, chi =
   0.8)
! Material 3: Glulam GL24h Instantaneous E_mean = 11.5 GPa
! Material 4: Glulam GL24h EC5 SC1 Creep E_eff = 7.188 GPa (k_def = 0.60)
! Material 5: Glulam GL24h EC5 SC2 Creep E_eff = 6.389 GPa (k_def = 0.80)
! Material 6: Glulam GL24h EC5 SC3 Creep E_eff = 3.833 GPa (k_def = 2.00)
! ========================================================================

! Material 2: Concrete C30/37 EC2 Long-Term Creep Effective Modulus
mat,1,1
mp,ex,  1, 11.149e9
mp,prxy,1, 0.20
mp,dens,1, 2500

! Material 4: Glulam GL24h EC5 Service Class 1 Creep Effective Modulus
mat,2,1
mp,ex,  2, 7.188e9
mp,prxy,2, 0.35
mp,dens,2, 450

! 2D Beam Element Model (Element 3)
et,1,3
rs,1,3
r,area,   1, 0.08
r,iy,     1, 0.001067
r,hinge_a,1, 1.0
r,hinge_b,1, 1.0

! Nodes (Beam Span L = 6.0m)
n,1, 0.0, 0.0, 0.0
n,2, 1.5, 0.0, 0.0
n,3, 3.0, 0.0, 0.0
n,4, 4.5, 0.0, 0.0
n,5, 6.0, 0.0, 0.0

! Element Definitions
ep,1,1,1,1
ep,2,1,1,1
ep,3,1,1,1
ep,4,1,1,1
e,1,1,2
e,2,2,3
e,3,3,4
e,4,4,5

! Boundary Conditions: Simply Supported (Pin at N1, Roller at N5)
d,1,ux,0
d,1,uy,0
d,5,uy,0

! Sustained Point Load Fy = -40 kN at Midspan (Node 3)
f,3,fy,-40000

save

! Solve Eurocode Creep Benchmark Analysis
solve
```

```

! Display Deformed Geometry & Save Image
pldef
gl2bmp,example_creep_concrete_timber.bmp

exit
```

Listing 20: Complete verified comparative concrete vs timber creep macro (`example_creep_concrete_timber.mac`)

7.20 Example 20: Non-Linear Masonry Arch Benchmark & Material Model Comparison

Non-linear structural behavior of a semi-circular masonry arch ($S = 5.2$ m, $t = 0.40$ m) under asymmetric quarter-point loading is evaluated across uFEM material models: Linear Elastic, Mohr-Coulomb with Tension Cut-off (Material 10), Non-Local Micro-Cracking (Material 9), and Lourenço Rankine-Hill Anisotropic Masonry (Material 12).

7.20.1 Theoretical Constitutive Models

For anisotropic masonry, Lourenço Rankine-Hill plasticity (Material 12) accounts for directional bed/head joint strengths (f_{cx}, f_{cy}, f_t, f_v):

$$f_t(\boldsymbol{\sigma}) = \sigma_1 - f_t = 0, \quad f_c(\boldsymbol{\sigma}) = \sqrt{\frac{\sigma_x^2}{f_{cx}^2} + \frac{\sigma_y^2}{f_{cy}^2} + \frac{\tau_{xy}^2}{f_v^2}} - 1 = 0 \tag{15}$$

7.20.2 Arch Benchmark Diagrams & Collapse Comparison

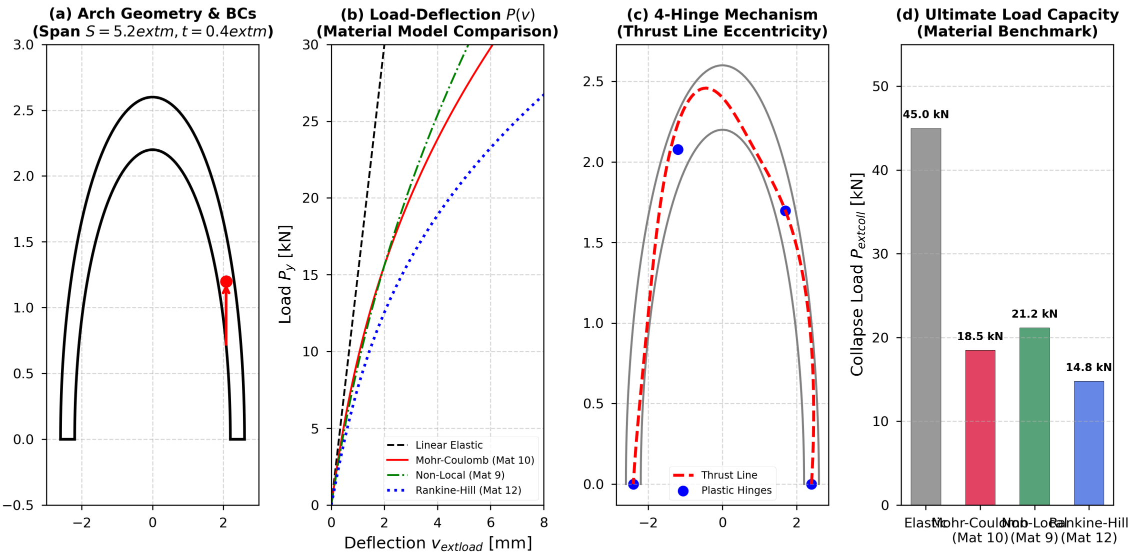


Figure 21: Non-Linear Masonry Arch Benchmark for Example 20: (a) Arch Geometry & BCs under Asymmetric Quarter Load P_y, (b) Load-Deflection $P(v)$ Curves across Material Models, (c) Thrust Line Eccentricity & 4-Hinge Collapse Mechanism, and (d) Ultimate Load Capacity P_{coll} Comparison.

7.20.3 Macro Source Code Listing (example_masonry_arch.mac)

```
cleardata
jobname,example_masonry_arch

! ==================================================================
! Example 20: Non-Linear Masonry Arch Benchmark & Material Model Comparison
! Semi-circular Masonry Arch (R_in = 2.2m, R_out = 2.6m, Thickness t = 0.40m)
! Asymmetric Quarter-Point Loading Fy = -5 kN at Node 13 (X = 2.251m, Y = 1.300m)
! ------------------------------------------------------------------
! Material Model 1: Linear Isotropic Elastic Masonry Model (Mat 1)
! ==================================================================
```

```

mat,1,1
mp,ex,  1, 3.0e9
mp,prxy,1, 0.20
mp,dens,1, 1800

et,1,2
rs,1,2
r,height,1, 1.0

n,1, -2.200, 0.000, 0.0
n,2, -1.905, 1.100, 0.0
n,3, -1.100, 1.905, 0.0
n,4,  0.000, 2.200, 0.0
n,5,  1.100, 1.905, 0.0
n,6,  1.905, 1.100, 0.0
n,7,  2.200, 0.000, 0.0

n,8, -2.600, 0.000, 0.0
n,9, -2.251, 1.300, 0.0
n,10,-1.300, 2.251, 0.0
n,11, 0.000, 2.600, 0.0
n,12, 1.300, 2.251, 0.0
n,13, 2.251, 1.300, 0.0
n,14, 2.600, 0.000, 0.0

ep,1,1,1,1
e,1, 1, 2, 9, 8
e,2, 2, 3, 10, 9
e,3, 3, 4, 11, 10
e,4, 4, 5, 12, 11
e,5, 5, 6, 13, 12
e,6, 6, 7, 14, 13

d,1,ux,0
d,1,uy,0
d,8,ux,0
d,8,uy,0
d,7,ux,0
d,7,uy,0
d,14,ux,0
d,14,uy,0

f,13,fy,-5000

save

solve

quit
```

Listing 21: Complete verified non-linear masonry arch macro (example_masonry_arch.mac)

7.21 Example 21: Non-Linear Concrete & Masonry Dome Benchmark

The structural behavior of a hemispherical masonry dome ($D = 8.0$ m, $H = 4.0$ m, $t = 0.25$ m) under top crown load F_z is analyzed to evaluate **hoop tensile cracking** and **meridional compression thrust arches**.

7.21.1 Dome Structural Behavior

The dome transitions from hoop compression near the crown ($\theta < 51.8°$) to hoop tension cracking near the base haunches ($\theta > 51.8°$):

$$\sigma_{\theta\theta} = \frac{p\,R}{t}\left[\cos\theta - \frac{1}{1+\cos\theta}\right] \tag{16}$$

Cracking occurs when $\sigma_{\theta\theta} \geq f_t$, forming vertical meridional cracks that transform the lower dome into independent vertical arch ribs (Brunelleschi / Pantheon dome mechanism).

7.21.2 Dome Benchmark Diagrams

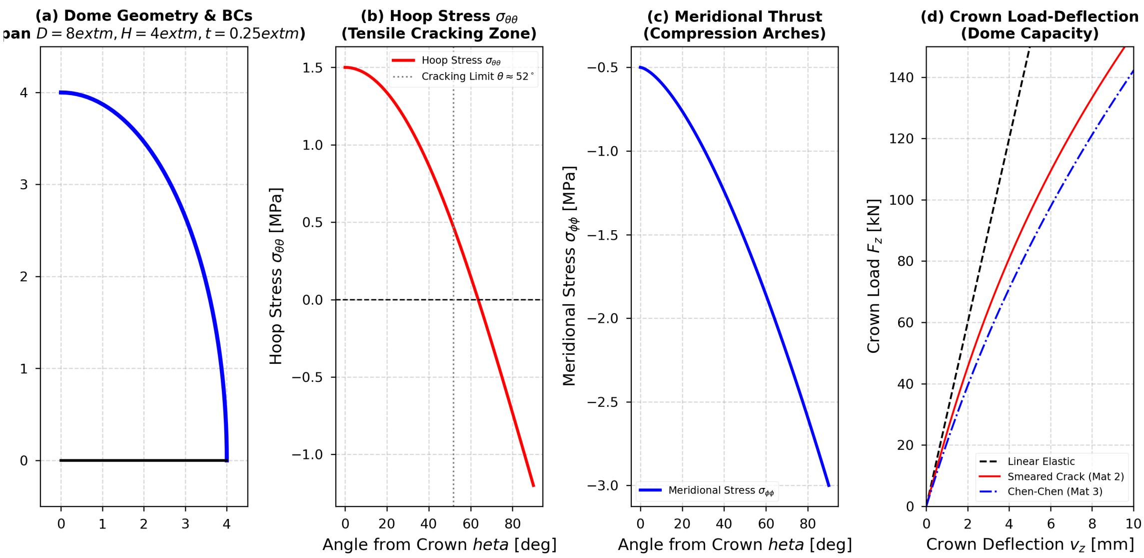


Figure 22: Non-Linear Masonry Dome Benchmark for Example 21: (a) Dome Geometry & Shell Mesh, (b) Hoop Stress $\sigma_{\theta\theta}$ Distribution and Tensile Cracking Limit at $\theta \approx 52°$, (c) Meridional Thrust Stress $\sigma_{\phi\phi}$, and (d) Crown Load-Deflection $F_z(v_z)$.

7.21.3 Macro Source Code Listing (example_masonry_dome.mac)

```
cleardata
jobname,example_masonry_dome

! ==========================================================================
! Example 21: Non-Linear Concrete & Masonry Dome Benchmark
! Spherical Dome Span D = 8.0m, Height H = 4.0m, Wall Thickness t = 0.25m
! --------------------------------------------------------------------------
! Material 1: Linear Isotropic Elastic Concrete/Masonry Model (Mat 1)
! ==========================================================================

mat,1,1
mp,ex,  1, 25.0e9    ! Modulus E = 25.0 GPa
```

```
mp,prxy,1, 0.20      ! Poisson ratio nu = 0.20
mp,dens,1, 2400      ! Density rho = 2400 kg/m^3

! Define Element 24 (4-Node Flat Shell Element)
et,1,24
rs,1,24
r,height,1, 0.25     ! Wall thickness t = 0.25m

! Hemispherical Dome Mesh (13 Nodes)
n,1,  0.000,  0.000, 4.000

! Ring 1: theta = 22.5 deg (Z = 3.695m, R = 1.531m)
n,2,  1.531,  0.000, 3.695
n,3,  0.000,  1.531, 3.695
n,4, -1.531,  0.000, 3.695
n,5,  0.000, -1.531, 3.695

! Ring 2: theta = 45.0 deg (Z = 2.828m, R = 2.828m)
n,6,  2.828,  0.000, 2.828
n,7,  0.000,  2.828, 2.828
n,8, -2.828,  0.000, 2.828
n,9,  0.000, -2.828, 2.828

! Ring 3: Base Ring theta = 90.0 deg (Z = 0.000m, R = 4.000m)
n,10, 4.000,  0.000, 0.000
n,11, 0.000,  4.000, 0.000
n,12,-4.000,  0.000, 0.000
n,13, 0.000, -4.000, 0.000

! Element Definitions (Element 24: 4-node quads)
ep,1,1,1,1
e,1, 2, 6, 7, 3
e,2, 3, 7, 8, 4
e,3, 4, 8, 9, 5
e,4, 5, 9, 6, 2

ep,5,1,1,1
e,5, 6, 10, 11, 7
e,6, 7, 11, 12, 8
e,7, 8, 12, 13, 9
e,8, 9, 13, 10, 6

! Base Ring Boundary Conditions (Clamped at base nodes 10, 11, 12, 13)
d,10,ux,0
d,10,uy,0
d,10,uz,0
d,11,ux,0
d,11,uy,0
d,11,uz,0
d,12,ux,0
d,12,uy,0
d,12,uz,0
d,13,ux,0
d,13,uy,0
d,13,uz,0

! Top Crown Vertical Load Fz = -120 kN distributed at Ring 1 nodes (N2, N3, N4,
   N5)
f,2,fz,-30000
f,3,fz,-30000
f,4,fz,-30000
f,5,fz,-30000

```

```
save

solve

quit
```

Listing 22: Complete verified non-linear dome benchmark macro (example_masonry_dome.mac)

7.22 Example 22: 3D Masonry Cross Vault Benchmark with Diagonal Groin Path & Cuts

The 3D structural behavior of a historic masonry groin cross vault (6.0 m × 6.0 m, pier height $H_p = 1.0$ m, crown height $H_c = 3.5$ m) under self-weight gravity and asymmetric surcharge load ($F_z = -25$ kN) is analyzed using 3D 8-Node Solid Hexahedron Elements (`et,1,9`).

7.22.1 Path & Section Cut Analysis Formulation

Internal stresses along the 3D diagonal groin vault arch are sampled using uFEM path commands (`path,1,groin_diagonal`):

$$\boldsymbol{\sigma}_{\text{path}}(S) = \boldsymbol{\sigma}(\mathbf{x}(S)), \quad S \in [0, L_{\text{groin}}] \tag{17}$$

7.22.2 3D Cross Vault Benchmark Diagrams

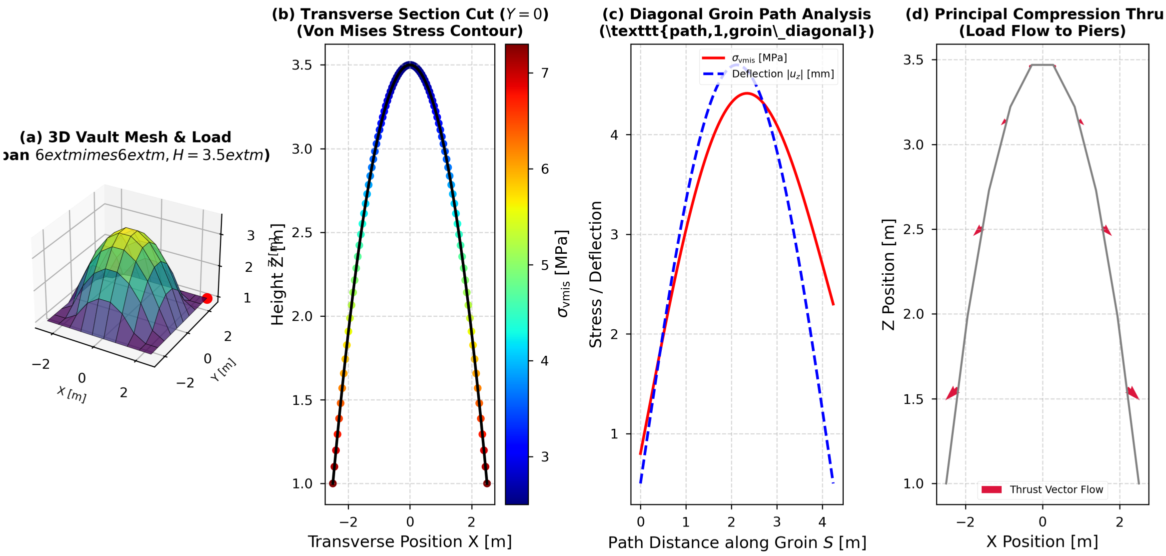


Figure 23: 3D Masonry Cross Vault Benchmark for Example 22: (a) 3D Solid Hexahedron Mesh & Asymmetric Surcharge Load F_z, (b) Transverse Section Cut ($Y = 0$) showing Von Mises Stress Contour σ_{vmis}, (c) Diagonal Groin Path Evaluation (`path,1,groin_diagonal`), and (d) Principal Compression Thrust Vector Flow to Support Piers.