AlfaCAD 2.7 Supplement 5. Gaining degrees of freedom.

In a static analysis context, "gaining degrees of freedom" would be a misnomer, as static analysis aims to solve for equilibrium by restricting degrees of freedom through supports and boundary conditions, not by adding them. In mechanics and structural analysis, degrees of freedom represent the number of ways an object can move or rotate, and static analysis determines the system's response by reducing these possible movements.

By the title of this article, the author refers to the ability to shape connections between elements of static systems by adding degrees of freedom relative to the state when all of them were previously taken away. If the elements are rigidly connected, but we add a degree of freedom for rotation at each node, we create a pinned connection, commonly used in truss structures.

What's the difference between a frame and a truss? Is there such a thing as a hybrid, part frame, part truss? Let's ask Grok:
(Grok is a generative artificial intelligence chatbot developed by xAI. It was launched in November 2023 by Elon Musk as an initiative based on the large language model (LLM) of the same name. Who knows everything about the differences between frames and trusses can skip this text in blue)

“Hi Grok. What would be the most accurate explanation what is the difference between frame and truss in static analyses ?”

In static structural analysis, frames and trusses are both skeletal structures used to support loads, but they differ in their design, behavior, and analysis. Here's a concise and accurate explanation of their differences:

1. Definition and Purpose
- *Truss: A truss is a structure composed of straight members connected at joints (typically pinned) to form a series of triangles. Trusses are designed to carry axial loads (tension or compression) only, with joints assumed to transfer no bending moments.*
- *Frame: A frame is a structure made of members (beams, columns, etc.) connected rigidly or semi-rigidly, designed to resist axial, shear, and bending moment forces. Frames are used to support a wider variety of loads and configurations, often in buildings or bridges.*

2. Member Behavior
- *Truss: Members are subjected to axial forces only (tension or compression). The pinned joints prevent bending moments, simplifying the analysis to axial load distribution.*
- *Frame: Members experience axial forces, shear forces, and bending moments. The connections (rigid or semi-rigid) allow moments to develop, making the analysis more complex.*

3. Joint Connections
- *Truss: Joints are typically pinned (idealized as frictionless), allowing rotation but no moment transfer. This assumption simplifies calculations but limits trusses to triangular geometries for stability.*
- *Frame: Joints are usually rigid or semi-rigid, transferring moments, shear, and axial forces. This allows frames to have more complex geometries, like rectangles, but requires analyzing moment distribution.*

4. Structural Geometry
- *Truss: Typically composed of triangular units, which are inherently stable and rigid. Common in lightweight structures like bridges, roofs, or towers.*
- *Frame: Can have various geometries (rectangular, triangular, etc.), as rigidity comes from the connections and member properties. Used in buildings, portals, or multi-story structures.*

5. Static Analysis Approach
- *Truss:*
- *Analyzed using methods like the method of joints or method of sections, focusing on equilibrium of forces ($\Sigma F = 0$) at each joint.*
- *Assumes members are two-force members (axial loads only).*
- *Simpler calculations due to the absence of moments.*
- *Frame:*
- *Analyzed using methods like moment distribution, slope-deflection, or matrix methods, accounting for axial, shear, and moment forces.*
- *Requires solving for internal forces and moments, often using structural analysis software for complex frames.*
- *More computationally intensive due to multiple force types.*

6. Examples
- *Truss: Roof trusses, bridge trusses (e.g., Warren or Howe truss), or crane booms.*
- *Frame: Building skeletons, portal frames, or multi-story structures with rigid joints.*

Summary Table

Feature	*Truss*	*Frame*
Loads Carried	*Axial (tension/compression)*	*Axial, shear, bending moments*
Joint Type	*Pinned (no moment transfer)*	*Rigid/semi-rigid (moment transfer)*
Geometry	*Triangular (for stability)*	*Any (rectangular, etc.)*
Analysis	*Method of joints/sections*	*Moment distribution, slope-deflection*
Complexity	*Simpler (axial forces only)*	*More complex (multiple forces)*
Applications	*Bridges, roofs, towers*	*Buildings, portals, frames*

Key Takeaway
Trusses are specialized structures optimized for axial loads with pinned joints, making them simpler to analyze but limited in application. Frames are more versatile, handling multiple force types with rigid joints, but require more complex analysis. The choice depends on the structural requirements and load conditions.

So how we can classify the structure which contains both members, some of them connected rigidly and some of them pinned at one or both joints ?

A structure that contains both rigidly connected members (resisting axial, shear, and bending moments) and pinned members (primarily resisting axial forces) is classified as a hybrid structure or mixed structure in static analysis. These structures combine characteristics of both trusses and frames, and their analysis requires a tailored approach that accounts for the behavior of each member type and connection. Here's a detailed explanation of how to classify and analyze such structures:

Classification of a Hybrid Structure

1. Mixed Behavior:

- Pinned members (like truss elements) are treated as two-force members, carrying only axial forces (tension or compression). These are typically connected at pinned joints, allowing rotation but no moment transfer.

- Rigidly connected members (like frame elements) resist axial forces, shear forces, and bending moments, with connections that transfer moments (rigid or semi-rigid joints).

- The structure as a whole cannot be strictly classified as a truss or a frame but is considered a hybrid structure because it exhibits both truss-like and frame-like behavior.

2. Structural Configuration:

- The structure may include triangular elements (typical of trusses) for stability, combined with rectangular or other geometries (typical of frames) where rigid joints provide additional stiffness.

For example, a bridge might have a truss-like lattice for the main span (pinned joints) but include rigid connections at supports or for certain beams to handle complex loading.

3. Terminology:

- Such structures are often called truss-frame systems, composite structures, or hybrid frameworks in engineering literature.

- In practice, they may be referred to based on their application (e.g., "bridge with mixed connections" or "building frame with truss elements").

Analysis of Hybrid Structures

Analyzing a hybrid structure requires combining techniques used for trusses and frames, depending on the member and joint types. Here's how to approach it:

1. Identify Member and Joint Types:

- Pinned joints: Assume no moment transfer; analyze for axial forces using truss methods (method of joints or sections).

- Rigid joints: Account for moment,shear and axial forces using frame analysis methods (moment distribution,slope-deflection or stiffness matrix methods).

- Semi-rigid joints (if present): Model partial moment transfer, often requiring advanced analysis with defined joint stiffness.

2. Decompose the Structure:

- Break the structure into subcomponents:

- Truss-like portions: Analyze pinned members using equilibrium equations ($\Sigma F = 0$) at joints, assuming axial forces only.

- Frame-like portions: Analyze rigid members for axial, shear, and moment forces, considering joint rotations and deflections.

- Ensure compatibility at interfaces where truss-like and frame-like elements meet (e.g., displacements and forces must be consistent).

3. Static Determinacy:

- Check if the structure is statically determinate or indeterminate:

- For trusses: Use the formula ($m + r = 2j$) (where m is the number of members, r is the number of support reactions, and j is the number of joints).

- For frames or hybrid structures: Include additional equations for moment equilibrium at rigid joints. If indeterminate, use methods like force or displacement methods.

- Hybrid structures are often statically indeterminate due to rigid connections, requiring advanced techniques.

4. Analysis Methods:

- Method of Joints/Sections (for truss-like portions): Solve for axial forces in pinned members.

- Moment Distribution or Slope-Deflection (for frame-like portions): Calculate moments, shears, and axial forces in rigid members.

- Matrix Methods or Finite Element Analysis (FEA): For complex hybrid structures, use computational tools to model both pinned and rigid connections. Software like SAP2000, ANSYS, or STAAD.Pro is commonly used.

- Compatibility Conditions: Ensure that displacements and rotations at joints connecting truss and frame elements are compatible.

5. Load Path and Behavior:

- Pinned members (truss-like) will primarily transfer loads axially, contributing to overall stability.

- Rigid members (frame-like) will resist bending and shear, allowing the structure to handle diverse loads (e.g., lateral loads like wind or seismic forces).

- The interaction between pinned and rigid elements must be carefully modeled to avoid incorrect assumptions about load distribution.

Practical Examples

- Building with Truss Roof and Frame Columns: A warehouse might have a truss roof (pinned joints for lightweight axial load transfer) supported by rigid columns and beams (frame-like) to resist lateral loads.

- Bridges with Mixed Systems: Some bridges combine truss elements for the main span (e.g., a Warren truss) with rigid portal frames at the ends to handle complex support conditions.

- Industrial Structures: Cranes or towers may use truss-like lattices for lightweight spans and rigid frames for support towers or bases.

Challenges in Analysis

- Complexity: Hybrid structures are often statically indeterminate, requiring advanced computational tools for accurate analysis.

- Joint Modeling: The transition between pinned and rigid joints must be carefully modeled to ensure accurate load transfer.

- Software Reliance: For practical design, engineers typically use FEA software to handle the mixed behavior, as manual calculations become cumbersome.

Summary

A structure with both pinned and rigid members is classified as a hybrid structure or truss-frame system. It combines truss-like behavior (axial forces in pinned members) and frame-like behavior (axial, shear, and moment forces in rigid members). Analysis involves separating truss and frame components, applying appropriate methods (e.g., method of joints for trusses, moment distribution for frames), and ensuring compatibility at interfaces. For complex cases, computational tools like FEA are recommended.

If you have a specific structure or diagram in mind, I can help you break down its analysis further! Would you like to provide more details or explore a specific example?

OK, that was Grok (3), used only for testing, and I think the test passed. We'll be using Grok again later, but for much more serious tasks, not just to confirm what we already know. However, when considering hybrid structures, I wasn't just referring to structures in which one part is a truss and another a frame, but rather to structures in which each element can gaining a degree of freedom at either end, and that degree of freedom is rotation around the local Z or Y axis, or both.

AlfaCAD is using the individual implementation of Frame3DD program, developed by Henri P. Gavin, Ph.D., P.E., Department of Civil and Environmental Engineering, Duke University, Durham, NC, U.S., and John Pye, Dept. of Engineering, Australian National University.

The changes and enhancements introduced in Frame3DD by AlfaCAD's author (including some corrections of non-critical errors) were dictated by the collaboration with AlfaCAD, both for the purpose of entering the geometry and characteristics of structural elements, as well as for system loads and their combinations, and for the interpretation and presentation of calculation results in both static and dynamic analysis.

Although the program allows for taking into account geometric stiffness, the influence of shear deformations, as well as the influence of infinite node stiffness within a given radius in the calculations, the biggest limitation of the program is the fact, that all connections in a Frame3DD analysis are moment-resisting. In original Frame3DD program manual is written:

"Internal hinges may be modeled using a short element with low values of *Jxx*, *Iyy*, and *Izz*. Many connections are more realistically modeled as having some flexibility. Such semi-rigid connections may be modeled through the inclusion of short frame elements with appropriate section and material properties to model the behavior of the connection. Frame elements may be considered infinitely rigid within a sphere of a specified radius, *r* around a node. The effects of finite node sizes are modeled approximately in the calculation of frame element stiffness through the use of an *effective beam length*, which is the node-to-node length of the frame element less the rigid radii on each end.

To analyze a structure as a "truss" with this software, specify *Jxx*, *Iyy*, and *Izz* to be much smaller than they would be normally, but not zero. If the shear forces and bending moments in the structural elements are small, then the structural model represents a "truss" approximation of the actual structure. Shear deformation effects and geometric stiffness effects should **not** be incorporated if *Jxx*, *Iyy*, and *Izz* are made very small. "

(https://svn.code.sourceforge.net/p/frame3dd/code/trunk/doc/Frame3DD-manual.html chapter: 7.7 Connections)

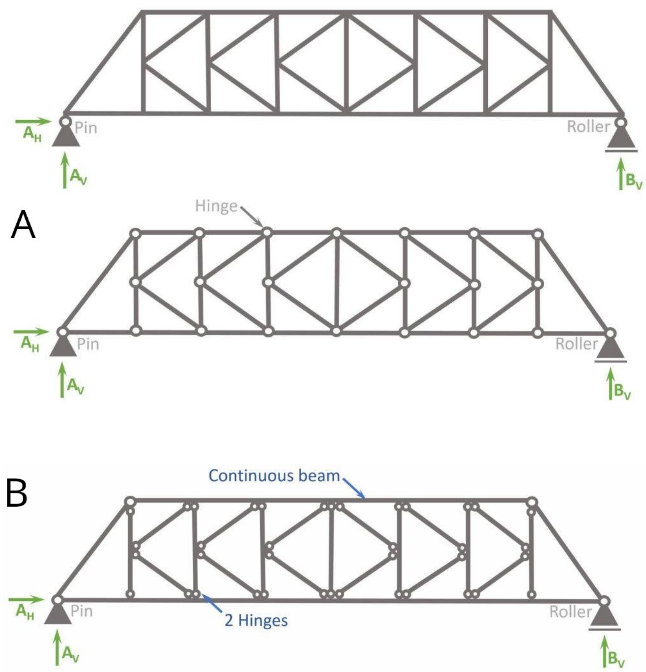
As a matter of facts, defining the finite elasticity of a connection between a member and other elements at a node is highly questionable. Such stiffness can be specified in absolute values in SI units of kNm/rad, or as a relative stiffness relative to the stiffness of the entire element, e.g., 50%. In both cases, this is difficult to assess, except for nodes specifically shaped for specific technological purposes or due to the requirements for the structure's resistance to dynamic effects from seismic loads.

In static analysis, we are always guided by certain assumptions that allow for the application of one theory or another. If we ensure joint rigidity in a frame structure through welded connections that prevent rotation of one element relative to another at the connection point, or use other connections (riveted, bolted) using gusset plates that allow for the same conclusion, then such a connection should be considered rigid. 100% rigid.

If we use a riveted or bolted connection in which the arrangement of the rivets or bolts does not guarantee that both elements are capable of relative rotation under load, or if we intentionally use a pinned connection (due to their design), then it should be considered pinned. 100% pinned.

This doesn't mean that the entire structure contains rigid or pinned connections. However, as long as all connections can be treated as rigid, entering static system data, calculating them using Frame3DD, and interpreting the calculation results is straightforward. When a connection should be considered a hinge, either because of its low bending stiffness or because its intentional design meets all the criteria for a pure hinge, both data entry and result interpretation become somewhat more complicated. However, AlfaCAD has so far handled this problem by, in accordance with the Frame3DD authors' suggestion, introducing very short elements with very low bending stiffness as hinges, while maintaining the same axial and shear stiffness as the entire element.

The photo on the right serves as an example.
It can be assumed with negligible error that the diagonals of each X-type crossing in each segment of the bridge span, running from the right node connecting the diagonal with the bottom chord and the left node connecting the element with the top chord, constitute a continuous beam. Therefore the connection of the two halves of this element at the middle node constitutes a rigid connection. Elements joining at the middle node, connected with rivets via joint plates, are de facto connected in a pinned manner, because such a connection does not ensure a complete absence of rotation and therefore the transfer of the bending moment.
The bottom and top chords of the truss should be treated as continuous beams, therefore, the connections between their segments should be rigid-rigid type.



Similarly, the truss diagram on the left, originally presented as a set of rigidly connected members, can be treated as a statically determinate system in which all connections are hinges, allowing a degree of rotational freedom at each node, as in diagram A, or, more closely resembling the real system, as a statically indeterminate system with continuous bottom and top chords, as in diagram B.

The speed and ease of performing static calculations using the program allows for the calculation of all three variants of the diagram, and the resulting envelopes of maximum cross-sectional forces can be used as a basis for designing the elements' cross-sections. However, such variants are not necessary if we use node types that have a technical justification. Sections of continuous chords should be rigidly connected, therefore the elements constituting them will be rigid-rigid vectors (in AlfaCAD nomenclature).

The remaining elements should adopt the form indicated by the technical specifications of the nodes, so if virtual rotation of the element end at the node is possible, that end should be hinged (or in another word - pinned).
This situation can occur, for example, in bolted or riveted connections, which do not provide complete stiffness because the distance of the bolts or rivets from the potential axis of rotation is small compared to the element's cross-sectional height, despite the use of gusset plates that increase the stiffness of the joint itself. In each case, the choice of static scheme belongs to the designer, and the program's task is to enable such selection without any compromise.

To incorporate true hinged-rigid, rigid-hinged, or hinged-hinged elements in addition to rigid-rigid elements, several significant changes and extensions must be made to the static and dynamic analysis software. These extensions can be divided into six groups:
1. Extensions to the static system stiffness matrix
2. Extensions to the system load vectors
3. Extensions to the system mass matrix
4. Extensions to the analysis of internal forces in elements based on nodal displacements
5. Extensions to the analysis of deformation of elements based on nodal displacements and internal forces
6. Extensions to the analysis of element deformations in vibration modes

It should be noted that a node where more than two elements meet, and not all elements are connected to the node in the same way, creates a situation where we are actually dealing with a multi-node, where only some of them have restraining of rotational degree of freedom, while others are free to rotate. So, if we have three elements, one of which is connected by a pin while the others are connected to a node rigidly, the node will rotate under the influence of loads from the rigidly mounted elements, but this will not affect the rotation of the virtual node of the pinned element. Therefore, if we imagine three virtual nodes belonging to each element, the equations assume that the rotations of the nodes of the rigidly mounted elements are identical when the rotation of the node of the pinned element is different, while the displacements of all virtual nodes in each axis (X, Y, Z) are identical.

However, the stiffness matrix contains a single node common to all three elements. The difference is that when constructing the stiffness matrix, we ignore the influence of the pinned element on the rotation of the node, and the influence on the remaining displacements takes into account the method of fastening the element at both ends. In the analysis of internal forces and deformation of elements, it should be taken into account that the rotation of the node constituting the pinned connection of a given element does not affect the internal forces or the deformation of the element, because this rotation, if different from zero, comes from the influence of the remaining elements, if such are connected in a rigid way.

A similar situation occurs with mass matrices in dynamic analysis. The vibrating masses of an element connected by a hinged node do not affect the rotation of that node.

Let's begin with the element stiffness matrix.

Here is the 3D elastic stiffness matrix for frame elements including shear and bending effects as used in Frame3DD:

$$\mathbf{k}_E = \begin{bmatrix}
\frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3(1+\Phi_y)} & 0 & 0 & 0 & \frac{6EI_z}{L^2(1+\Phi_y)} & 0 & \frac{-12EI_z}{L^3(1+\Phi_y)} & 0 & 0 & 0 & \frac{6EI_z}{L^2(1+\Phi_y)} \\
0 & 0 & \frac{12EI_y}{L^3(1+\Phi_z)} & 0 & \frac{-6EI_y}{L^2(1+\Phi_z)} & 0 & 0 & 0 & \frac{-12EI_y}{L^3(1+\Phi_z)} & 0 & \frac{-6EI_y}{L^2(1+\Phi_z)} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{L^2(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)EI_y}{L(1+\Phi_z)} & 0 & 0 & 0 & \frac{6EI_y}{L^2(1+\Phi_z)} & 0 & \frac{(2-\Phi_z)EI_y}{L(1+\Phi_z)} & 0 \\
0 & \frac{6EI_z}{L^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{(4+\Phi_y)EI_z}{L(1+\Phi_y)} & 0 & \frac{-6EI_z}{L^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{(2-\Phi_y)EI_z}{L(1+\Phi_y)} \\
\frac{-EA_x}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA_x}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-12EI_z}{L^3(1+\Phi_y)} & 0 & 0 & 0 & \frac{-6EI_z}{L^2(1+\Phi_y)} & 0 & \frac{12EI_z}{L^3(1+\Phi_y)} & 0 & 0 & 0 & \frac{-6EI_z}{L^2(1+\Phi_y)} \\
0 & 0 & \frac{-12EI_y}{L^3(1+\Phi_z)} & 0 & \frac{6EI_y}{L^2(1+\Phi_z)} & 0 & 0 & 0 & \frac{12EI_y}{L^3(1+\Phi_z)} & 0 & \frac{6EI_y}{L^2(1+\Phi_z)} & 0 \\
0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{L^2(1+\Phi_z)} & 0 & \frac{(2-\Phi_z)EI_y}{L(1+\Phi_z)} & 0 & 0 & 0 & \frac{6EI_y}{L^2(1+\Phi_z)} & 0 & \frac{(4+\Phi_z)EI_y}{L(1+\Phi_z)} & 0 \\
0 & \frac{6EI_z}{L^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{(2-\Phi_y)EI_z}{L(1+\Phi_y)} & 0 & \frac{-6EI_z}{L^2(1+\Phi_y)} & 0 & 0 & 0 & \frac{(4+\Phi_y)EI_z}{L(1+\Phi_y)}
\end{bmatrix},$$

where

$$\Phi_y = \frac{12EI_z}{GA_{sy}L^2}, \qquad \text{and} \qquad \Phi_z = \frac{12EI_y}{GA_{sz}L^2}.$$

If we momentarily ignore the effects of shear and bending (geometric stiffness effects) to simplify our considerations, the stiffness matrix looks like this:

$$\begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ M_{x,1} \\ M_{y,1} \\ M_{z,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \\ M_{x,2} \\ M_{y,2} \\ M_{z,2} \end{Bmatrix} = \begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & 0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\
0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 & 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\
\frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} & 0 & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} \\
0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_z}{L^2} & 0 \\
0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & \frac{-6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\
0 & \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & 0 & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L}
\end{bmatrix} \begin{Bmatrix} U_{x,1} \\ U_{y,1} \\ U_{z,1} \\ \varphi_{x,1} \\ \varphi_{y,1} \\ \varphi_{z,1} \\ U_{x,2} \\ U_{y,2} \\ U_{z,2} \\ \varphi_{x,2} \\ \varphi_{y,2} \\ \varphi_{z,2} \end{Bmatrix}$$

AlfaCAD is a 2D CAD program, and the static systems it solves are also 2D, although the calculations are performed in 3D, taking into account loads in the XY plane and ignoring loads in the XZ plane.

Therefore, by shaping a joint as pinned, we mean unrestrained rotational degree of freedom for rotation about the local Z axis of the element at one or both of its ends, leaving restrained rotational degrees of freedom for rotation about the Y axis and X axis (torsion). In fact, we won't change $M_{x,1}$ $M_{y,1}$ $M_{x,2}$ and $M_{y,2}$ as well as $\varphi_{x,1}$ $\varphi_{y,1}$, $\varphi_{x,2}$ and $\varphi_{y,2}$. That can be done in the future for 3D systems as well, in a similar way.

$$
\begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ M_{x,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \\ M_{x,2} \\ M_{y,2} \\ M_{z,2} \end{Bmatrix} =
\begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{3EI_z}{L^3} & 0 & 0 & 0 & \frac{-3EI_z}{L^3} & 0 & 0 & 0 & \frac{3EI_z}{L^2} \\
0 & 0 & \frac{3EI_y}{L^3} & 0 & 0 & 0 & \frac{-3EI_y}{L^3} & 0 & \frac{-3EI_y}{L^2} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\
\frac{-EA}{L} & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{-3EI_z}{L^3} & 0 & 0 & 0 & \frac{3EI_z}{L^3} & 0 & 0 & 0 & \frac{-3EI_z}{L^2} \\
0 & 0 & \frac{-3EI_y}{L^3} & 0 & 0 & 0 & \frac{3EI_y}{L^3} & 0 & \frac{3EI_z}{L^2} & 0 \\
0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\
0 & 0 & \frac{-3EI_y}{L^2} & 0 & 0 & 0 & \frac{3EI_y}{L^2} & 0 & \frac{3EI_y}{L} & 0 \\
0 & \frac{3EI_z}{L^2} & 0 & 0 & 0 & \frac{-3EI_z}{L^2} & 0 & 0 & 0 & \frac{3EI_z}{L}
\end{bmatrix}
\begin{Bmatrix} U_{x,1} \\ U_{y,1} \\ U_{z,1} \\ \varphi_{x,1} \\ U_{x,2} \\ U_{y,2} \\ U_{z,2} \\ \varphi_{x,2} \\ \varphi_{y,2} \\ \varphi_{z,2} \end{Bmatrix}
$$

And here is the form of the stiffness matrix of the element whose end (1) is hinged (here, for generalization, it is assumed that the connection is hinged both with respect to rotation around the local Z axis and the local Y axis; only rotation around the local X axis is restrained). Hence, $M_{y,1}$ and $M_{z,1}$, as well as $\varphi_{y,1}$ and $\varphi_{z,1}$ do not occur (are equal zero), to demonstrate the solution for 3D systems.

$$
\begin{Bmatrix} F_{x,1} \\ F_{y,1} \\ F_{z,1} \\ M_{x,1} \\ M_{y,1} \\ M_{z,1} \\ F_{x,2} \\ F_{y,2} \\ F_{z,2} \\ M_{x,2} \end{Bmatrix} =
\begin{bmatrix}
\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-EA}{L} & 0 & 0 & 0 \\
0 & \frac{3EI_z}{L^3} & 0 & 0 & 0 & \frac{3EI_z}{L^2} & 0 & \frac{-3EI_z}{L^3} & 0 & 0 \\
0 & 0 & \frac{3EI_y}{L^3} & 0 & \frac{-3EI_y}{L^2} & 0 & 0 & 0 & \frac{-3EI_y}{L^3} & 0 \\
0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{-GJ}{L} \\
0 & 0 & \frac{-3EI_y}{L^2} & 0 & \frac{3EI_y}{L} & 0 & 0 & 0 & \frac{3EI_y}{L^2} & 0 \\
0 & \frac{3EI_z}{L^2} & 0 & 0 & 0 & \frac{3EI_z}{L} & 0 & \frac{-3EI_z}{L^2} & 0 & 0 \\
\frac{-EA}{L} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{L} & 0 & 0 & 0 \\
0 & \frac{-3EI_z}{L^3} & 0 & 0 & 0 & \frac{-3EI_z}{L^2} & 0 & \frac{3EI_z}{L^3} & 0 & 0 \\
0 & 0 & \frac{-3EI_y}{L^3} & 0 & \frac{3EI_y}{L^2} & 0 & 0 & 0 & \frac{3EI_y}{L^3} & 0 \\
0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & 0 & 0 & \frac{GJ}{L}
\end{bmatrix}
\begin{Bmatrix} U_{x,1} \\ U_{y,1} \\ U_{z,1} \\ \varphi_{x,1} \\ \varphi_{y,1} \\ \varphi_{z,1} \\ U_{x,2} \\ U_{y,2} \\ U_{z,2} \\ \varphi_{x,2} \end{Bmatrix}
$$

Here is the form of the stiffness matrix of the element whose end (2) is hinged.

$$
\begin{Bmatrix} F_{x,1} \\ M_{x,1} \\ F_{x,2} \\ M_{x,2} \end{Bmatrix} =
\begin{bmatrix}
\frac{EA}{L} & 0 & \frac{-EA}{L} & 0 \\
0 & \frac{GJ}{L} & 0 & \frac{-GJ}{L} \\
\frac{-EA}{L} & 0 & \frac{EA}{L} & 0 \\
0 & \frac{-GJ}{L} & 0 & \frac{GJ}{L}
\end{bmatrix}
\begin{Bmatrix} U_{x,1} \\ \varphi_{x,1} \\ U_{x,2} \\ \varphi_{x,2} \end{Bmatrix}
$$

Here is the form of the stiffness matrix of the element whose both ends are hinged. The elements only contribute to axial forces and node displacements along the local X axis, as well as torsional moments and rotation about the local X axis.

These matrices are declared in the procedure `elastic_K(…)` in the file `frame3dd.c`

Assembling load vectors

Load information, originally as load vectors expressed in local coordinates, has to be assembled as load vectors in global coordinates, in the form of vector of equivalent nodal forces F_temp and F_mech and a matrix of equivalent element end forces eqF_temp and eqF_mech from distributed internal and temperature loadings. eqF_temp and eqF_mech are computed for the global coordinate system.

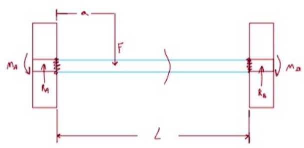
In AlfaCAD / Frame3DD we have type of loads:

- gravity loads applied uniformly to all frame elements
- node point loads
- uniformly distributed loads
- trapezoidally distributed loads
- internal element point loads
- thermal loads
- prescribed displacements of nodes

Elements and loads

Let’s take a sheet of paper (it’s too hard with text editor) and let's try to derive fixed end moments and forces of a member fixed at both ends, with a point force applied in the distance “a” from left support, as an exercise how to do that analytically:

1)

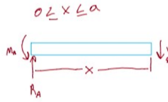


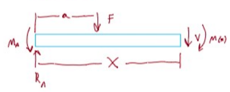
$\Sigma F_y = 0 = R_A + R_B - F \quad (1)$

$\Sigma M_F: \; -M_B + M_A + R_B L - Fa = 0 \quad (2)$

translational

$0 \le x \le a$ $\qquad\qquad$ $a \le x \le L$





$\Sigma F_y: \; V = R_A$ $\qquad$ $\Sigma F_y; \; V = -F + R_A$

$\Sigma M_x \; m(x) = M_A - R_A x$ $\qquad$ $\Sigma M_x \; m(x) = M_A + F(x-a) - R_A(x)$

$$\Delta = \int_0^a \frac{M_x\left(\frac{\partial m(x)}{\partial R_A}\right) dx}{EI} \quad + \quad \Delta = \int_a^L \frac{M_x\left(\frac{\partial m(x)}{\partial R_A}\right) dx}{EI}$$

$$\frac{\partial m(x)}{\partial R_A} = -x \qquad\qquad \frac{\partial m(x)}{\partial R_A} = -x$$

$$\frac{1}{EI}\int_0^a (M_A - R_A x)(-x)\,dx \quad + \quad \frac{1}{EI}\int_a^L (M_A + F(x-a) - R_A x)(-x)\,dx$$

$$\frac{1}{EI}\int_0^a (-M_A x + R_A x^2)\,dx \quad + \quad \frac{1}{EI}\int_a^L (-M_A x - Fx^2 + Fax + R_A x^2)\,dx$$

$$\frac{1}{EI}\left[\frac{-M_A x^2}{2} + \frac{R_A x^3}{3}\right]\Bigg|_0^a \quad + \quad \frac{1}{EI}\left[\frac{-M_A x^2}{2} - \frac{Fx^3}{3} + \frac{Fax^2}{2} + \frac{R_A x^3}{3}\right]\Bigg|_a^L$$

$$\Delta = \frac{1}{EI}\left(\cancel{-\frac{M_A a^2}{2}} + \cancel{\frac{R_A a^3}{3}} - \frac{M_A L^2}{2} - \frac{FL^3}{3} + \frac{FaL^2}{2} + \frac{R_A L^3}{3} + \cancel{\frac{M_A a^2}{2}} + \frac{Fa^3}{3} - \frac{Fa^3}{2} - \cancel{\frac{R_A a^3}{3}}\right)$$

$$\Delta = \frac{1}{EI}\left(-\frac{M_A L^2}{2} - \frac{FL^3}{3} + \frac{FaL^2}{2} + \frac{R_A L^3}{3} + \frac{Fa^3}{3} - \frac{Fa^3}{2}\right) \quad (3)$$

$$M_A = -\frac{2FL}{3} + Fa + \frac{2R_A L}{3} + \frac{2Fa^3}{3L^2} - \frac{Fa^3}{L^2}$$

3) $$M_A = -\frac{2FL}{3} + Fa + \frac{2\left(F - \frac{3Fa^2}{L^2} + \frac{2Fa^3}{L^3}\right)L}{3} + \frac{2Fa^3}{3L^2} - \frac{Fa^3}{L^2}$$

$$M_A = -\cancel{\frac{2FL}{3}} + Fa + \cancel{\frac{2FL}{3}} - \frac{2Fa^2}{L} + \frac{4Fa^3}{3L^2} + \frac{2Fa^3}{3L^2} - \frac{Fa^3}{L^2}$$

$$\boxed{M_A = Fa - \frac{2Fa^2}{L} + \frac{Fa^3}{L^2}}$$

$\Sigma F_y = 0 = R_A + R_B - F \quad (1)$

$$R_B = -\left(\cancel{F} - \frac{3Fa^2}{L^2} + \frac{2Fa^3}{L^3}\right) \cancel{+F}$$

$$\boxed{R_B = \frac{3Fa^2}{L^2} - \frac{2Fa^3}{L^3}}$$

$\Sigma M_F: \; -M_B + M_A + R_B L - Fa = 0 \quad (2)$

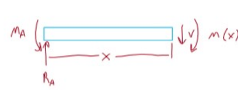
$M_B = M_A + R_B L - Fa$

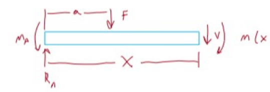
$$M_B = \cancel{Fa} - \frac{2Fa^2}{L} + \frac{Fa^3}{L^2} + \frac{3Fa^2}{L} - \frac{2Fa^3}{L^2} - \cancel{Fa}$$

$$\boxed{M_B = \frac{Fa^2}{L} - \frac{Fa^3}{L^2}}$$

2) rotational

$0 \le x \le a$ $\qquad\qquad$ $a \le x \le L$





$\Sigma F_y: \; V = R_A$ $\qquad$ $\Sigma F_y; \; V = -F + R_A$

$\Sigma M_x = 0 = M_A - R_A(x)$ $\qquad$ $\Sigma M_x = 0 = M_A + F(x-a) - R_A(x)$

$$\Delta = \int_0^a \frac{M_x\left(\frac{\partial m(x)}{\partial M_A}\right) dx}{EI} \quad + \quad \Delta = \int_a^L \frac{M_x\left(\frac{\partial m(x)}{\partial M_A}\right) dx}{EI}$$

$$\frac{\partial m(x)}{\partial M_A} = 1 \qquad\qquad \frac{\partial m(x)}{\partial M_A} = 1$$

$$\frac{1}{EI}\int_0^a (M_A - R_A x)\,dx \quad + \quad \frac{1}{EI}\int_a^L (M_A + Fx - Fa - R_A x)\,dx$$

$$\frac{1}{EI}\left[M_A x - \frac{R_A x^2}{2}\right]\Bigg|_0^a \quad + \quad \frac{1}{EI}\left[M_A x + \frac{Fx^2}{2} - Fax - \frac{R_A x^2}{2}\right]\Bigg|_a^L$$

$$\Delta = \frac{1}{EI}\left(\cancel{M_A a} - \cancel{\frac{R_A a^2}{2}} \quad + \quad M_A L + \frac{FL^2}{2} - FaL - \frac{R_A L^2}{2} - \cancel{M_A a} - \frac{Fa^2}{2} + Fa^2 + \cancel{\frac{R_A a^2}{2}}\right)$$

$$\Delta = \frac{1}{EI}\left(M_A L + \frac{FL^2}{2} - FaL - \frac{R_A L^2}{2} - \frac{Fa^2}{2} + Fa^2\right) \quad (4)$$

$$R_A = \frac{2M_A}{L} + F - \frac{2Fa}{L} - \frac{Fa^2}{L^2} + \frac{2Fa^2}{L^2}$$

$$R_A = \frac{2\left(-\frac{2FL}{3} + Fa + \frac{2R_A L}{3} + \frac{2Fa^3}{3L^2} - \frac{Fa^3}{L^2}\right)}{L} + F - \frac{2Fa}{L} - \frac{Fa^2}{L^2} + \frac{2Fa^2}{L^2}$$

$$R_A = -\frac{4F}{3} + \cancel{\frac{2Fa}{L}} + \frac{4R_A}{3} + \frac{4Fa^3}{3L^3} - \frac{2Fa^3}{L^3} + F - \cancel{\frac{2Fa}{L}} - \frac{Fa^2}{L^2} + \frac{2Fa^2}{L^2}$$

$$-\frac{R_A}{3} = -\frac{4F}{3} + \frac{4Fa^3}{3L^3} - \frac{2Fa^3}{L^3} + F - \frac{Fa^2}{L^2} + \frac{2Fa^2}{L^2}$$

$$\cancel{-3} \cdot \frac{R_A}{\cancel{3}} = -3\,\frac{F}{3} - 3\,\frac{2Fa^3}{3L^3} + {}^{-3}\frac{Fa^2}{L^2}$$

$$\boxed{R_A = F - \frac{3Fa^2}{L^2} + \frac{2Fa^3}{L^3}}$$

Let’s confront results with ready to use formulas from “The Engineering ToolBox” : https://www.engineeringtoolbox.com/beams-fixed-both-ends-support-loads-deflection-d_809.html

Bending Moment

$M_A = - F\, a\, b^2 / L^2$

$M_B = - F\, a^2\, b / L^2$

Support Reactions

$R_A = F\, (3\, a + b)\, b^2 / L^3$

$R_B = F\, (a + 3\, b)\, a^2 / L^3$

Let’s check if equations found do match the equations above, and they are just written differently but will yield the same outcome:

so as a test let’s check if $R_{B\,(\text{found})} = R_{B\,(\textit{from toolbox})}$:

$3Fa^2/L^2 – 2Fa^3/L^3 = F\,(a + 3\,b)\,a^2 / L^3$ $\qquad$ $3Fa^2/L^2 – 2Fa^3/L^3 = (Fa^3 + F3ba^2) / L^3$

$3Fa^2/L^2 – 2Fa^3/L^3 = Fa^3/ L^3 + F3ba^2 / L^3$ $\qquad$ $3Fa^2/L^2 – 2Fa^3/L^3 - Fa^3/ L^3 + F3ba^2 / L^3 = 0$

$3Fa^2/L^2 – 3Fa^3/L^3 + F3ba^2 / L^3 = 0$ $\qquad$ let’s multiply by L^3

$3Fa^2L – 3Fa^3 + F3ba^2 = 0$ $\qquad$ $3Fa^2\,(L – a – b) = 0$ lets divide by $3Fa^2$

$L – a – b = 0$ so $\qquad$ $L = a + b$ what is TRUTH

Now let’s check if $M_{B\,(\text{found})} = M_{B\,(\textit{from toolbox})}$:

$– (Fa^2/L – Fa^3/L^2) = – F\,a^2\,b / L^2$ (we changed sign of found moment due to moments sign convention) let’s multiply by L^2

$– (Fa^2L – Fa^3) + F\,a^2\,b = 0$ let’s multiply by -1

$(Fa^2L – Fa^3) – F\,a^2\,b = 0$

$Fa^2\,(L – a – b) = 0$ let’s divide by Fa^2

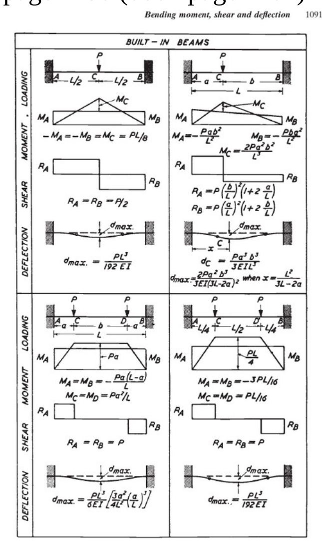
$L – a – b = 0$ so $\qquad$ $L = a + b$ what is TRUTH

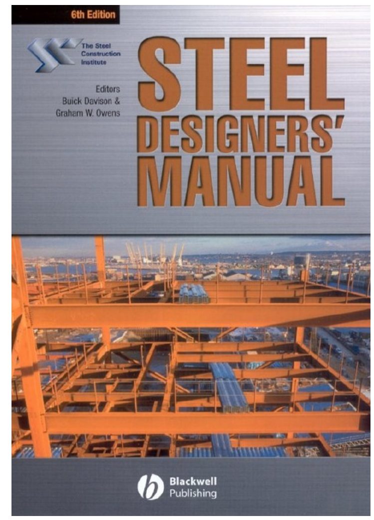
It takes time and substantial effort to derive all formulas for fixed member, propped cantilever where alternative fixed end, as well as simply supported (pinned) element, for all types of load (concentrated, uniformly distributed, trapezoidally distributed, eventually thermal load). All of those are already done, so it's easier to use existing formulas, taken for example from such Engineering ToolBox like this:

https://www.engineeringtoolbox.com/beams-fixed-end-d_560.html
or from this great book:
https://www.uceb.eu/DATA/Books/Steel%20Designer%27s%20Manual,%206th%20Edition%20-%20(Malestrom).pdf
(pages: from PDF page 1125 (book page 1077) to PDF page 1150 (book page 1101)

Some variables here:

If shear deformation is taken into account
Ksz = (12.0*E[n]*Iy[n]) / (G[n]*Asz[n]*Le[n]*Le[n]);
otherwise Ksz = 0;
    where Ksz is shear deformation coefficients
E[n] elastic moduli of element n
G[n] shear moduli of element n
Iy [n] section inertia of element n
Asz [n] section area of element n
Ln or L[n] the length of n frame element
Le or Le[n] the effective length of n frame element
P is the concentrated point load
U is uniformly distributed load
W is trapezoidally distributed load
T is temperature load





Node point loads

The type of element connections does not affect the load distribution.

Uniformly distributed loads (including gravity loads applied uniformly to all frame elements, if the gravitational acceleration value is set)

Fixed-Fixed element
node reactions are:

Vy1 = Vy2 = U * L[n] / 2.0;
Mz1 = U * L[n] * L[n] / 12.0;
Mz2 = -Mz1;

Pinned-Fixed element
mode reactions are:

Vy1 = 3. * U * L[n] / 8.0;
Vy2 = 5. * U * L[n] / 8.0;
Mz1 = 0.0;
Mz2 = -U * L[n] * L[n] / 8.0;

Fixed-pinned element
node reactions are:

Vy1 = 5. * U * L[n] / 8.0;
Vy2 = 3. * U * L[n] / 8.0;

Mz1 = U * L[n] * L[n] / 8.0;
Mz2 = 0.0;

Pinned-pinned element
node reactions are:

Vy1 = Vy2 = U * L[n] / 2.0;
Mz1 = Mz2 = 0.0;

Trapezoidally distributed load

Fixed-Fixed element
node reactions are:

explanation of indices:

lc is the number of load case
i is a number of load (can be multiple within one element)
j is a data index in vector:

	X-axis	Y-axis	Z-axis
1 element number	2 start x-coordinate	6 start x-coordinate	10 start x-coordinate
	3 stop x-coordinate	7 stop x-coordinate	11 stop x-coordinate
	4 start magnitude	8 start magnitude	12 start magnitude
	5 stop magnitude	9 stop magnitude	13 stop magnitude

Only indices 6 .. 9 are to calculate Vy1, Vy2 and Mz1, Mz2 which are different for different types of members

```
/* y-axis trapezoidal loads (across the frame element length) */

x1 =  W[lc][i][6];  x2 = W[lc][i][7];

w1 =  W[lc][i][8]; w2 =  W[lc][i][9];

R1o = ( (2.0*w1+w2)*x1*x1 - (w1+2.0*w2)*x2*x2 + 3.0*(w1+w2)*Ln*(x2-x1) - (w1-w2)*x1*x2 ) / (6.0*Ln);

R2o = ( (w1+2.0*w2)*x2*x2 + (w1-w2)*x1*x2 - (2.0*w1+w2)*x1*x1 ) / (6.0*Ln);

f01 = (  3.0*(w2+4.0*w1)*x1*x1*x1*x1 -  3.0*(w1+4.0*w2)*x2*x2*x2*x2
        - 15.0*(w2+3.0*w1)*Ln*x1*x1*x1 + 15.0*(w1+3.0*w2)*Ln*x2*x2*x2
        -  3.0*(w1-w2)*x1*x2*(x1*x1 + x2*x2)
        + 20.0*(w2+2.0*w1)*Ln*Ln*x1*x1 - 20.0*(w1+2.0*w2)*Ln*Ln*x2*x2
        + 15.0*(w1-w2)*Ln*x1*x2*(x1+x2)
        -  3.0*(w1-w2)*x1*x1*x2*x2 - 20.0*(w1-w2)*Ln*Ln*x1*x2 ) / 360.0;

f02 = (  3.0*(w2+4.0*w1)*x1*x1*x1*x1 - 3.0*(w1+4.0*w2)*x2*x2*x2*x2
        -  3.0*(w1-w2)*x1*x2*(x1*x1+x2*x2)
        - 10.0*(w2+2.0*w1)*Ln*Ln*x1*x1 + 10.0*(w1+2.0*w2)*Ln*Ln*x2*x2
        -  3.0*(w1-w2)*x1*x1*x2*x2 + 10.0*(w1-w2)*Ln*Ln*x1*x2 ) / 360.0;
```

hence:

Mz1 = -(4.0*f01 + 2.0*f02 + Ksy*(f01 - f02)) / (Ln*Ln*(1.0+Ksy));
Mz2 = -(2.0*f01 + 4.0*f02 - Ksy*(f01 - f02)) / (Ln*Ln*(1.0+Ksy));
Vy1 = R1o + Mz1/Ln + Mz2/Ln;
Vy2 = R2o - Mz1/Ln – Mz2/Ln;

Pinned-Fixed element
mode reactions are:

```
////counted analytically
aa=x1;
bb=Ln-x2;
Lw=Ln-bb-aa;
wm=(w2+w1)/2.;
wd=w1-w2;
s1=40.*Ln*(2.*Ln*Ln-Lw*Lw)+10.*Lw*(Lw*Lw-2.*bb*bb)-40.*bb*(Ln-aa)*(2.*Ln+aa);
s2=Lw*(3.*Lw*Lw-10.*Ln*(Lw+2.*bb)+10.*bb*(bb+Lw));
```

hence:

Vy2=Lw*(s1*wm+s2*wd)/(80.*Ln*Ln*Ln);
Vy1=Lw*wm-Vy2;
Mz1=0.0;
Mz2=Vy1*Ln-bb*Lw*wm-Lw*Lw*(2.*w1+w2)/6.;

(in this case shear deformation effect was neglected)

Alternatively, it could be also calculated (including shear deformation) using numerical integration, in the form:

```
int ni=100;  //number of segments
double dxi=Lw/ni;
double dPi=(w2-w1)/ni;
double Pni;  //partial force for integration
double ai, bi;  //force distances from nodes
int in;

Vy1=0.0;
Vy2=0.0;
Mz1=0.0;
Mz2=0.0;

for (in=0; in<ni; in++)
{
   Pni=(w1+(in+1)*dPi + w1+(in)*dPi)*dxi/2.;
   //Pni=w1+(in+0.5)*dPi*dxi;
   ai=x1+((in+1)*dxi + (in)*dxi)/2.;
   //ai=x1+(in+0.5)*dxi;
   bi=Ln-ai;

   Vy1 += Pni*bi*bi*(ai+(2.*Ln))/(2.*Ln*Ln*Ln) + (Ksz/(1.+Ksz)) * Pni*bi/Ln;
   Vy2 += Pni*ai*(3.*Ln*Ln-ai*ai)/(2.*Ln*Ln*Ln)  //// 20-07-2025 + (Ksz/(1.+Ksz)) * Pni*ai/Ln;
   Mz2 += -(1. / (1. + Ksz)) * Pni * (ai*ai*bi + (bi*bi*ai/2.))/ (Ln * Ln) - (Ksz / (1. + Ksz)) * Pni * ai * bi / (2. * Ln);
}
```

Fixed-pinned element

node reactions are:

////counted analytically
aa=x1;
bb=Ln-x2;
Lw=Ln-aa-bb;
wm=(w1+w2)/2.;
wd=w2-w1;
s1=40.*Ln*(2.*Ln*Ln-Lw*Lw)+10.*Lw*(Lw*Lw-2.*aa*aa)-40.*aa*(Ln-bb)*(2.*Ln+bb);
s2=Lw*(3.*Lw*Lw-10.*Ln*(Lw+2.*aa)+10.*aa*(aa+Lw));

Vy1=Lw*(s1*wm+s2*wd)/(80.*Ln*Ln*Ln);
Vy2=Lw*wm-Vy1;
Mz1= - (Vy2*Ln-aa*Lw*wm-Lw*Lw*(2.*w2+w1)/6.);
Mz2=0.0;

Alternatively, it could be also calculated (including shear deformation) using numerical integration, in the form:

```
int ni=100;  //number of segments
double dxi=Lw/ni;
double dPi=(w2-w1)/ni;
double Pni;  //partial force for integration
double ai, bi;  //force distances from nodes
int in;

Vy1=0.0;
Vy2=0.0;
Mz1=0.0;
Mz2=0.0;

for (in=0; in<ni; in++) {
   Pni=(w1+(in+1)*dPi + w1+(in)*dPi)*dxi/2.;
   //Pni = w1 + (in + 0.5) * dPi * dxi;
```

```
        ai=x1+((in+1)*dxi + (in)*dxi)/2.;
        //ai = x1 + (in + 0.5) * dxi;
        bi = Ln - ai;

        Vy1 += Pni*bi*(3.*Ln*Ln-bi*bi)/(2.*Ln*Ln*Ln) + (Ksz/(1.+Ksz)) * Pni*bi/Ln;
        Vy2 += Pni*ai*ai*(bi+(2.*Ln))/(2.*Ln*Ln*Ln) + (Ksz/(1.+Ksz)) * Pni*ai/Ln;
        Mz1 += (1. / (1. + Ksz)) * Pni * (bi*bi*ai + (ai*ai*bi/2.))/ (Ln * Ln) - (Ksz / (1. + Ksz)) * Pni * ai * bi / (2. * Ln);
    }
```

Pinned-pinned element
node reactions are:

```
P1=(x2-x1)*w1;
P2=(x2-x1)*(w2-w1)/2.;
a1=(x2-x1)/2.+x1;
b1=Ln-a1;
a2=0.0;
b2=0.0;
if (w2!=w1)  {
    a2=(fabs(w2)>fabs(w1)) ? 2.*(x2-x1)/3.+x1 : (x2-x1)/3.+x1;
    b2=Ln-a2;  }
```

Mz1 = **Mz2** = 0.0;
Vy1 = P1*b1/Ln + (Ksz/(1.+Ksz)) * P1*b1/Ln; //// first substitute force
Vy1 += P2*b2/Ln + (Ksz/(1.+Ksz)) * P2*b2/Ln; //// second substitute force

Vy2 = P1*a1/Ln + (Ksz/(1.+Ksz)) * P1*a1/Ln;
Vy2 += P2*a2/Ln + (Ksz/(1.+Ksz)) * P2*a2/Ln;

Internal element point load

Fixed-Fixed element
node reactions are:

Vy1 = (1./(1.+Ksz))   * P * b * b * (3.* a + b) / (Ln * Ln * Ln) + (Ksz/(1.+Ksz)) * P * b/Ln;
Vy2 = (1./(1.+Ksz))   * P * a * a *(3.* b + a) / (Ln * Ln *Ln) + (Ksz/(1.+Ksz)) * P * a/Ln;
Mz1 = (1. / (1. + Ksz)) * P * a * b * b / (Ln * Ln) + (Ksz / (1. + Ksz)) * P * a * b / (2. * Ln);
Mz2 = -(1. / (1. + Ksz)) * P * a * a * b / (Ln * Ln) - (Ksz / (1. + Ksz)) * P * a * b / (2. * Ln);

Pinned-Fixed element
mode reactions are:

Vy1 = P * b * b * (a+(2. * Ln))/(2. * Ln * Ln * Ln) + (Ksz/(1.+Ksz)) * P * b/Ln;
Vy2 = P * a * (3.* Ln * Ln - a * a)/(2. * Ln * Ln * Ln) + (Ksz/(1.+Ksz)) * P * a/Ln;
Mz1 = 0.0;
Mz2 = - (1. / (1. + Ksz)) * P * (a *a * b + (b * b * a/2.))/ (Ln * Ln) - (Ksz / (1. + Ksz)) * P * a * b / (2. * Ln);

Fixed-pinned element
node reactions are:

Vy1 = P * b * (3.* Ln * Ln – b * b)/(2. * Ln * Ln * Ln) + (Ksz/(1.+Ksz)) * P * b/Ln;
Vy2 = P * a * a * (b+(2.* Ln))/(2.* Ln * Ln * Ln) + (Ksz/(1.+Ksz)) * P * a/Ln;
Mz1 = (1. / (1. + Ksz)) * P * (b * b * a + (a * a * b/2.))/ (Ln * Ln) - (Ksz / (1. + Ksz)) * P * a * b / (2. * Ln);
Mz2 = 0.0;

Pinned-pinned element
node reactions are:

Vy1 = P * b/Ln + (Ksz/(1.+Ksz)) * P * b/Ln;
Vy2 = P * a/Ln + (Ksz/(1.+Ksz)) * P * a/Ln;
Mz1 = Mz2 = 0.0;

Thermal loads

indices:
lc is load case number
i is i is a number of load (can be multiple within one element)
j is a data index in vector:

1 element number	3 y-depth	5 deltaTy+	7 deltaTz+
2 coefficient of thermal expansion	4 z-depth	6 deltaTy-	8 deltaTz-

a = T[lc][i][2];
hy = T[lc][i][3];
hz = T[lc][i][4];

Fixed-Fixed element
node reactions are:

Vy1 = **Vy2** = 0.0;
Mz1 = (a / hy) * (T[lc][i][5] - T[lc][i][6]) * E[n] * Iz[n];
Mz2 = -Mz1;

Pinned-Fixed element
mode reactions are:

tm=(a / hy) * (T[lc][i][5] - T[lc][i][6]) * E[n] * Iz[n];

Vy1= -3.*tm/L[n]/2.;
Vy2= 3.*tm/L[n]/2.;
Mz1 = 0.0;
Mz2 = -3.*tm/2.;

Fixed-pinned element
node reactions are:
tm=(a / hy) * (T[lc][i][5] - T[lc][i][6]) * E[n] * Iz[n];

Vy1= 3.*tm/L[n]/2.;

Vy2= -3.*tm/L[n]/2.;
Mz1 = 3.*tm/2.;
Mz2 = 0.0;

Pinned-pinned element
node reactions are:

Vy1 = **Vy2** = 0.0;
Mz1 = **Mz2** = 0.0;

Those formulas are inserted in procedure `read_and_assemble_loads(…)` in the file `frame3dd_io.c`

AI incident

In the mid-2020s, we gained new sources of information, even knowledge, and above all, substantive assistance in solving problems, including technical ones. For example, let's compare some formulas with what Grok, considered one of the best AI models, has to say on the subject. In this case, Grok 3, although we know that Grok 4 and even 5 already exist. Chatting with Grok (transcript is attached below), I asked about formulas for support reactions in an element (beam) fixed on both sides, under the load of a concentrated force. These reactions are equivalent to the forces on the system nodes connecting the element, which is what we need when defining the load matrix.

As you can see in the attached transcript, as long as the AI model is able to "copy" ready-to-use formulas from sources available in published documents, assuming these formulas are correct, the model answers the question correctly. This concerns the reaction in an element constrained on both sides (page 6 of the chat).
My next question concerned an element simply supported on the left side (in the local coordinate system) and fixed on the opposite side (propped cantilever).
The solution generated by AI (page 10) was clearly different from the solution I arrived at by reversing the formulas for the opposite system, it means the element fixed on the left and simply supported on the right. Recognizing this temporary as an error, I reversed the question, asking for formulas that could be "downloaded from the web" (page 11 of the chat).
Despite the AI's use of "Standard Reaction Formulas," the obtained results do not satisfy the equilibrium conditions (page 15 of the chat), as I have demonstrated. The next solution (page 20) left me perplexed...
So I suggested the formulas found on www.engineeringtoolbox.com.
Ultimately, the AI arrived at a solution (page 24) identical to the one I obtained from the smart book (my bravado was limited to deriving formulas for a rigid-rigid members, and for only one type of load, a concentrated force; the remaining formulas were taken from the book).

So I returned to the previous question, which was the only one relevant to me because the task was to confirm the correctness of the formulas I derived based on the beam system with inverted supports.

Inverting the problem, that is the solution of an element simply supported on the left and fixed on the right , a somewhat simple system, a solution for which I also had trouble finding in the web resources and in the book, presents significant challenges for the AI.
The AI refers to the so-called "Standard Reaction Formulas," but the result is false and does not satisfy the basic boundary conditions, yet the model recognizes them as the "Final Answer" (page 29).
Interacting with the AI and pointing out the unsatisfied boundary conditions resulted in a different solution (page 34), not meeting the boundary conditions again, and after I pointed out this error Grok derived another “Final Solution” (page 42), but different, and in fact contradictory, to the one I derived by my own, basing on the solution for an element with inverted supports. However, Grok requested for my formulas for confirmation, and to ensure they are consistent or that there are any discrepancies. So I sent to him my solutions.
Eventually Grok stated that the validity of my formulas had been verified (page 47) and apologized for the confusion caused by incorrect standard formulas (whatever it "meant").

Okay, this is Grok-3. I've heard that Grok-4 is better and Grok-5 is the best. But I use Grok-3 because it's free and I don't want to buy a pig in a poke, so I don't use Grok-5.

The question arises: Can Grok-5 solve a problem that Grok-3 doesn't solve, or even solves without even knowing what it's talking about? As long as there are enough resources online, it will answer. Otherwise, it says: Give me your answer and I'll tell you if it's true. Could Grok-5 do that well ? If yes, why Grok-3 still exists ?
It's like going to a car dealer and they offer you a nice car for free, except it doesn't drive. But if you pay, they'll give you another one that does. So who needs the one that doesn't drive?

Waste of time. Maybe one day...
But later, I'll share my positive experience with Grok-3, though not without troubles.

Chat source: [Fixed Beam Support Reactions Analysis - Grok-3]

Solving the system

System is solved in the procedure `solve_system(…)` in the file `frame3dd.c` .
The result is the displacement values of all nodes for all degrees of freedom. It should be remembered that the rotation of a node declared as a hinge in a given element is not influenced by the loads on the element, nor is the rotation of this node (due to the loads of other elements connected to this node in a rigid manner) influencing the distribution of forces and moments in a given element.
The system is solved multiple times for all load combinations (for Eurocode standard even up to 61 cycles for all ULS and SLS combinations, bit less for ASCE and ICC standards), first for thermal loads (if any) and then for mechanical loads, after which the effects of both solutions are combined together.
Next, quasi Newton-Raphson iteration for geometric nonlinearity is performed, if the effect of geometric stiffness is set up.

Element forces {Q} for displacements {D}

Procedure `element_end_forces(…)` in the file `frame3dd.c` evaluates the member end forces for every member in the procedure `frame_element_force(…)` where extra variables are provided:
`int n1y, int n1z, int n2y, int n2z` which are the indices of the rotation restraint of node 1 relative to the Y axis (due to the 2D nature of the systems solved in AlfaCAD, this parameter always has the value 1, which means that the node is rigid in the YZ plane) and the Z axis (value 1 means rotation restraint, value 0 means free rotation) and node 2 relative to the Y axis and Z axis.

n1 and n2 are the numbers of element's nodes 1 and 2 in the system (counting originally from 1) so they are converted to indices of displacement vector D previously solved in solve_system(…) procedure.

```
n1 = 6*(n1-1);     n2 = 6*(n2-1);    // 6 is the number of degrees of freedom, that's why

d1  = D[n1+1];    d2  = D[n1+2];    d3  = D[n1+3];
d4  = D[n1+4];    d5  = D[n1+5];    d6  = D[n1+6];
d7  = D[n2+1];    d8  = D[n2+2];    d9  = D[n2+3];
d10 = D[n2+4];    d11 = D[n2+5];    d12 = D[n2+6];
```

So:
d1 is displacement of node 1 in local axis X
d2 is displacement of node 1 in local axis Y
d3 is displacement of node 1 in local axis Z (effectively negligible in 2D system)
d4 is the rotation of node 1 around the X axis (torsion, negligible in 2D system)
d5 is the rotation of node 1 around the Y axis (negligible in 2D system)
d6 is the rotation of node 1 around the Z axis

d7...d12 are the same but for node 2

s is a vector of forces, so:
s[1] is an axial force in the element at node 1
s[2] is a shear force in local axis Y of the element at node 1
s[3] is a shear force in local axis Z of the element at node 2 (negligible in 2D system)
s[4] is a moment about the X axis (torsion) in the element at node 1 (negligible in 2D system)
s[5] is a moment about the Y axis (bending) in the element at node 1 (negligible in 2D system)
s[6] is a moment about the Z axis (bending) in the element at node 1

s7...s12 are the same but for node 2

Le is effective length of the element (Le=L-r1-r2) where r1 is a radius of node 1 (if > 0) and r2 is a radius of node 2 (if > 0).
Ksy and **Dsy** are a shear deformation coefficients, and it is included, however, AlfaCAD ignores yet shear deformation effect (shear = 0), what is in most of cases negligible in 2D and 3D system, however in next edition it will be included too.

```
  if ( shear ) {
                Ksy = 12.*E*Iz / (G*Asy*Le*Le);
                Ksz = 12.*E*Iy / (G*Asz*Le*Le);  //(negligible in 2D system)
                Dsy = (1+Ksy)*(1+Ksy);
                Dsz = (1+Ksz)*(1+Ksz);  //(negligible in 2D system)
        }
        else {
                Ksy = Ksz = 0.0;
                Dsy = Dsz = 1.0;
        }
```

So now, taking into account the type of element, we get the formulas, by analogy to stiffness matrix element

Shear forces in node 1:

```
if ((n1z==1) && (n2z==1))  ////fixed-fixed
   {
      s[2] = -(12. * E * Iz / (Le * Le * Le * (1. + Ksy)) +  T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) *   ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             + (6. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *  ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else if ((n1z==0) && (n2z==1))  ///pinned-fixed
   {
      s[2] = -(3. * E * Iz / (Le * Le * Le * (1. + Ksy)) +  T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) *  ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             + (3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *   ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else if ((n1z==1) && (n2z==0))  ///fixed-pinned
   {
      s[2] = -(3. * E * Iz / (Le * Le * Le * (1. + Ksy)) +  T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) *  ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             + (3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *     ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else ////pinned-pinned
      s[2] = 0.0;
```

Now moments in node 1:

```
if ((n1z==1) && (n2z==1))  ////fixed-fixed
   {
     s[6] = -(6. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *  ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)  + ((4. + Ksy) * E * Iz / (Le * (1. + Ksy))
             + T * L * (2.0 / 15.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) *  (d4 * t7 + d5 * t8 + d6 * t9)
             + ((2. - Ksy) * E * Iz / (Le * (1. + Ksy)) -  T * L * (1.0 / 30.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) *  (d10 * t7 + d11 * t8 + d12 * t9);
   }
   else if ((n1z==0) && (n2z==1))  ///pinned-fixed
   {
      s[6] = 0.0;
   }
   else if ((n1z==1) && (n2z==0))  ///fixed-pinned
   {
      s[6] = -(3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) * ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             + ((3. + Ksy) * E * Iz / (Le * (1. + Ksy)) +  T * L * (2.0 / 15.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) *  (d4 * t7 + d5 * t8 + d6 * t9);
   }
   else  ////pinned-pinned
      s[6] = 0.0;
```

Now shear forces in node 2:

```
if ((n1z==1) && (n2z==1))  ////fixed-fixed
   {
      //s[8] = -s[2];   //or literally:

      s[8] =  (12. * E * Iz / (Le * Le * Le * (1. + Ksy)) + T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) * ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             - (6. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) * ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else if ((n1z==0) && (n2z==1))  ///pinned-fixed
   {
      s[8] =   (3. * E * Iz / (Le * Le * Le * (1. + Ksy)) + T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) * ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
             - (3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *  ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else if ((n1z==1) && (n2z==0))  ///fixed-pinned
   {
      s[8] =  (3. * E * Iz / (Le * Le * Le * (1. + Ksy)) + T / L * (1.2 + 2.0 * Ksy + Ksy * Ksy) / Dsy) * ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
               - (3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) * ((d4 + d10) * t7 + (d5 + d11) * t8 + (d6 + d12) * t9);
   }
   else ////pinned-pinned
      s[8] = 0.0;
```

and moments in node 2:

```
if ((n1z==1) && (n2z==1))  ////fixed-fixed
   {
    s[12] = -(6. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) *  ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6) + ((4. + Ksy) * E * Iz / (Le * (1. + Ksy))
             + T * L * (2.0 / 15.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) * (d10 * t7 + d11 * t8 + d12 * t9)
             + ((2. - Ksy) * E * Iz / (Le * (1. + Ksy)) -  T * L * (1.0 / 30.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) * (d4 * t7 + d5 * t8 + d6 * t9);
   }
   else if ((n1z==0) && (n2z==1))  ///pinned-fixed
```

```
{
   s[12] = -(3. * E * Iz / (Le * Le * (1. + Ksy)) + T / 10.0 / Dsy) * ((d7 - d1) * t4 + (d8 - d2) * t5 + (d9 - d3) * t6)
            + ((3. + Ksy) * E * Iz / (Le * (1. + Ksy)) + T * L * (2.0 / 15.0 + Ksy / 6.0 + Ksy * Ksy / 12.0) / Dsy) *  (d10 * t7 + d11 * t8 + d12 * t9);
}
else if ((n1z==1) && (n2z==0))  ///fixed-pinned
{
   s[12] = 0.0;
}
else ////pinned-pinned
   s[12] = 0.0;
```

Formulas for other forces are identical despite type of the member, so we skip presenting them here.
Deducing these all formulas is not difficult, as they directly relate to the relationships between forces and displacements expressed in the stiffness matrix of each type of the element of the system.

After obtaining the nodal forces, we need to determine the internal forces in the cross-sections of the elements and their deformations, i.e. displacements in the determined cross-sections.

Internal forces and displacements

Internal forces and displacements are calculated in procedure `write_internal_forces(…)` in file `frame3dd_io.c`
Let's look into the process focusing just on shear forces and moments, because they are different for different type of members:
For each cross-section, therefore for (i=1; i<=nx; i++), where nx is the number of cross-sections and x[i] is the distance of cross-section "i" from the first node along the local X axis, Nx[i], Vy[i], Vz[i], Tx[i], My[i], and Mz[i] are determined for each element load type. Let's focus on Nx[i], Vy[i], and Mz[i], which is only relevant for 2D systems. However, Nx[i] does not change depending on the element type.

Focusing on Nx, Vy and Mz (we ignore here Vz, Tx and My as not relevant for 2D systems)
where **m** is member number
0 is minimal number of cross section, so it's cross section in node 1
nx is a maximal number of cross section, so it's cross section in node 2

```
Nx[0] = -Q[m][1];            // positive Nx is tensile
Vy[0] = -Q[m][2];            // positive Vy in local y direction
////if fixed-fixed element or fixed-pin element
Mz[0] = -Q[m][6];            // positive Mz -> positive x-y curvature
```

additionally:
```
Mz[nx] = -Q[m][12];  //for pin-fixed  purpose, because we know Mz[0]=0;
```

In local x, y, z coordinates, for distributed gravity and uniform loads, trapezoidally distributed loads, interior point loads we do trapezoidal integration to get shear forces Vy[i] in each cross section.
Once we have Vy[i] we can find bending moments in each cross-section:
```
for (i = 1; i <= nx; i++)
{
     dx_ = x[i] - x[i - 1];
     Mz[i] = Mz[i - 1] - 0.5 * (Vy[i] + Vy[i - 1]) * dx_;
}
```
and now the integration correction, and – only for non rigid-rigid members, knowing Mzt[0] and Mzt[nx] (bending moments in element on its ends coming from thermal gradient load) we temporary subtract bending moments coming from thermal loads, because it would be disturbing in the process of calculating displacement for non rigid-rigid elements, due to the nature of thermal load (displacement of cross sections are not necessary related to bending moments in element; for example, simply supported beam under the thermal gradient load shows experience no bending moments, but shows cross section displacement (curvature)).
The distribution of bending moments from the thermal load between Mtz[0] and Mtz[nx] is always linear or has a constant value.
Thermal load moment subtraction is not performed for rigid-rigid elements. This is because we know the actual rotation of the element's nodes, and therefore there is a direct relationship between bending moments and the rotation of the nodes. This means that if the node rotation is restrained and the node is not free to rotate, the thermal gradient load generates a bending

moment. For non-rigid elements, the thermal gradient load does not generate a bending moment in the unrestrained node, but the element bends. Therefore, in members with a pinned joint or both pinned joints, the element deformations must be divided into those resulting from thermal loads and those resulting from mechanical loads.

```
for (i=0; i<=nx; i++)
{
        i_nx = x[i] / L[m];
        Mz[i] -= (Mz[nx] - Q[m][12]) * i_nx; //i/nx;

        //// removing thermal moment - it's just temporary
        if ((n1z[m] == 0) && (n2z[m] == 1)) Mz[i] -= (Mzt[nx]-Mzt[0]) * (i_nx);  ////for hinged-rigid
            else if ((n1z[m] == 1) && (n2z[m] == 0)) Mz[i] += (Mzt[nx]-Mzt[0]) * (1- i_nx);  ////for rigid-hinged

         ////it could be the same done for My[i] moments in 3D systems
}
```

Now we can calculate transverse slope along frame element "m" (here nodes radi was also taken into account, if > 0)

```
for (i=1; i<=nx; i++)
{
          dx_=x[i]-x[i-1];

          if (((r1==0) || (x[i]>r1)) && ((r2==0) || (x[i]<L[m]-r2)))
          {
                  Sy[i] = Sy[i - 1] + 0.5 * (Mz[i - 1] + Mz[i]) / (E[m] * Iz[m]) * dx_;
          }
          else
          {
                  Sy[i] = Sy[i - 1];
          }
}
```

and linear correction for bias in trapezoidal integration

```
double atansy=(Sy[nx] - u12);
for (i=1; i<=nx; i++)
{
        i_nx=x[i]/L[m];

        if ((n1z[m]==1) && (n2z[m]==1)) Sy[i] -= (Sy[nx] - u12) * i_nx;   //// rigid-rigid
        else if (n2z[m]==1)  //// pinned-rigid
           ;  //nada
        else if (n1z[m]==1)  //// rigged-pinned
           ;  //nada
        else   //// pinned-pinned
           ;   //nada

}
```

so there is no correction for non rigid-rigid members.

We can calculate displacement along frame element "m"

```
for (i=1; i<=nx; i++)
{
        dx_=x[i]-x[i-1];
        Dy[i] = Dy[i-1] + 0.5*(Sy[i-1]+Sy[i])*dx_;
}
```

and do linear correction for bias in trapezoidal integration

```
for (i=1; i<=nx; i++)
{
        i_nx=x[i]/L[m];
        Dy[i] -= (Dy[nx] - u8) * i_nx; //i/nx;
}
```

Extra displacement due to thermal load, then restore original cross section moments previously reduced due to thermal load This is done just for non rigid-rigid members.

```
if ((n1z[m]==0) && (n2z[m]==1)) ////hinged-rigid
      {
         for (i = 1; i < nx; i++)
         {
            Dy[i] += (Mzt[nx]/(6.*E[m]*Iz[m]*L[m]))*((x[i]*x[i]*x[i]) - (2.*x[i]*x[i]*L[m]) + (L[m]*L[m]*x[i]));
            ////identical calculation could be done for Dz[i] using Myt[nx]
         }
         for (i = 0; i <= nx; i++) {
            i_nx = x[i] / L[m];
            ////restoring thermal moment
            Mz[i] += (Mzt[nx] - Mzt[0]) * (i_nx);
            ////identical could be done for My[i], Myt[nx] and My[0]
         }

      }
      else if ((n2z[m]==0) && (n1z[m]==1)) ////rigid-hinged
      {
         for (i = 1; i < nx; i++)
         {
            Dy[i] -= (Mzt[0]/(6.*E[m]*Iz[m]*L[m]))*((x[i]*x[i]*x[i]) - (L[m]*x[i]*x[i]));
            ////identical calculation could be done for Dz[i] using Myt[0]
         }
         for (i = 0; i <= nx; i++) {
            i_nx = x[i] / L[m];
            ////restoring thermal moment
            Mz[i] -= (Mzt[nx]-Mzt[0]) * (1- i_nx);
            ////identical could be done for My[i], Myt[nx] and My[0]
         }
      }
      else if ((n1z[m]==0) && (n2z[m]==0)) ////hinged-hinged
      {
         if (Qt[m][13]!=0.0) {
            for (i = 1; i < nx; i++) {
              Dy[i] -= (Qt[m][13] * x[i] * (L[m] - x[i]));
            }
         }
            ////here is no thermal moment to restore; in pinned-pinned member the bending moment of thermal gradient is always equal 0
            ////identical calculation could be done for Dz[i] using Qt[m][14]
      }
```

We can assume that the calculation of the additional displacement due to the temperature gradient is correct, but the formulas (in bold in the code above) don't seem trivial. How do we derive them? Grok partially helped here, although without human assistance, none of this would have worked.

So after deriving formulas by my own based on my knowledge and ability to make math transformations, little bit generalizing on start I've asked Grok-3 first:
"what is the displacement of each point of propped cantilever beam, simply supported on left end and fixed on right end, if graph of internal moments in beam is linear from Ma at left end to Mb at right end".
Linear, because bending moment under thermal gradient load is always linear, but - what in fact I didn't mention - is linear on a specific way. In fact or bending moment is constant, or is linearly changing from zero to maximal value, or from maximal value to zero, depending of supports. So Ma=Mb, or Ma=0, or Mb=0, however, generalization was driven because we always deal with combination of thermal and mechanical loads.
The final answer (page 8 in the chat) is conditional, limiting the relation between M_A and M_B what was not a point.
Refining the question, claiming I meant thermal loading, I received a different answer (page 17).
Unfortunately, that solution didn't meet boundary conditions. So Grok tried again. The solution was close, but not exact (page 27). After auxiliary question about bending moment (page 28) and clarifying linearity of bending graph (page 35), then rejection of the generalization and focusing just on thermal load (page 37), I got proper solution (page 42) which I could numerically confirm on examples in AlfaCAD, modifying Frame3DDa procedures.

So I asked for inverted solution, propped cantilever with fixed on the left and simply supported on the right. Solution (page 49) not the first time didn't meet basic boundary conditions. It was guess rather than deriving formula. After pointing that, finally I got the "Final Answer" (page 64) which worked for me, quickly checking in AlfaCAD, because I knew what I have to get, so if the formula gave me expected result, I could say it's OK.
It took some time to solve the problem with the little help of the artificial intelligence. Ultimately, it succeeded, but the question remains: what value does artificial intelligence have without the human's one ?
This is what for such asked question Google AI answered:
Artificial intelligence (AI) can offer significant value through increased efficiency, automation, and data analysis, but it's fundamentally limited by its lack of human-like qualities such as creativity, emotional intelligence, and adaptability to novel situations. While AI can excel at repetitive tasks and complex calculations, it cannot replicate the nuanced understanding, critical thinking, and ethical reasoning that humans bring to problem-solving.

That's it.   Here is a transcript of the chat: Propped Cantilever Beam Deflection Analysis – Grok-3

Dynamic analysis

In the procedure `assemble_M(…)` in the file `frame3dd.c` is assembled the global mass matrix from element mass & inertia. There are 2 subroutines: `lumped_M(…)` for frame element lumped mass matrix, and `consistent_M(…)` for frame consistent mass matrix.

Values t1, t2, t3, t4, t5, t6, t7, t8, t9 are coordinates transformation factors, translating local coordinates system to global one for every element, so here is not necessary to care about them too much, they will just transform from local to global system, and they are calculated in `coord_trans(…)` procedure in the file `coordtrans.c`

Mass matrix, so the influence of **lumped** mass inertia for nodes movement for different types of element are the following:

```
/* rotatory inertia of extra mass is neglected */
t = ( d*Ax*L + EMs ) / 2.0;        /* translational inertia */
ry = d*Iy*L / 2.0;                 /* rotational inertia around Y axis */
rz = d*Iz*L / 2.0;                 /* rotational inertia around Z axis */
po = d*L*J / 2.0;                  /* polar inertia, assumes simple cross-section */

if ((n1z==1) && (n2z==1))  ////fixed-fixed
   {
      m[1][1] = m[2][2] = m[3][3] = m[7][7] = m[8][8] = m[9][9] = t;
      m[4][4] = m[10][10] = po * t1 * t1 + ry * t4 * t4 + rz * t7 * t7;
      m[5][5] = m[11][11] = po * t2 * t2 + ry * t5 * t5 + rz * t8 * t8;
      m[6][6] = m[12][12] = po * t3 * t3 + ry * t6 * t6 + rz * t9 * t9;
      m[4][5] = m[5][4] = m[10][11] = m[11][10] = po * t1 * t2 + ry * t4 * t5 + rz * t7 * t8;
      m[4][6] = m[6][4] = m[10][12] = m[12][10] = po * t1 * t3 + ry * t4 * t6 + rz * t7 * t9;
      m[5][6] = m[6][5] = m[11][12] = m[12][11] = po * t2 * t3 + ry * t5 * t6 + rz * t8 * t9;
   }
else if  ((n1z==0) && (n2z==1))  ////pinned-fixed
   {
      m[1][1] = m[2][2] = m[3][3] = m[7][7] = m[8][8] = m[9][9] = t;
      m[4][4] = m[10][10] = po * t1 * t1 + ry * t4 * t4 + rz * t7 * t7;
      m[5][5] = m[11][11] = po * t2 * t2 + ry * t5 * t5 + rz * t8 * t8;
      m[6][6] = 0;
      m[12][12] = po * t3 * t3 + ry * t6 * t6 + rz * t9 * t9;
      m[4][5] = m[5][4] = m[10][11] = m[11][10] = po * t1 * t2 + ry * t4 * t5 + rz * t7 * t8;
      m[4][6] = m[6][4] = 0;
      m[10][12] = m[12][10] = po * t1 * t3 + ry * t4 * t6 + rz * t7 * t9;
      m[5][6] = m[6][5] = 0;
      m[11][12] = m[12][11] = po * t2 * t3 + ry * t5 * t6 + rz * t8 * t9;
   }
else if  ((n1z==1) && (n2z==0))  ////fixed-pinned
   {
      m[1][1] = m[2][2] = m[3][3] = m[7][7] = m[8][8] = m[9][9] = t;
      m[4][4] = m[10][10] = po * t1 * t1 + ry * t4 * t4 + rz * t7 * t7;
      m[5][5] = m[11][11] = po * t2 * t2 + ry * t5 * t5 + rz * t8 * t8;
      m[6][6] = po * t3 * t3 + ry * t6 * t6 + rz * t9 * t9;
      m[12][12] = 0;
      m[4][5] = m[5][4] = m[10][11] = m[11][10] = po * t1 * t2 + ry * t4 * t5 + rz * t7 * t8;
      m[4][6] = m[6][4] = po * t1 * t3 + ry * t4 * t6 + rz * t7 * t9;
      m[10][12] = m[12][10] = 0;
```

m[5][6] = m[6][5] = po * t2 * t3 + ry * t5 * t6 + rz * t8 * t9;
m[11][12] = m[12][11] = 0;
}
else ////Pinned-Pinned
{
m[1][1] = m[2][2] = m[3][3] = m[7][7] = m[8][8] = m[9][9] = t;
m[4][4] = m[10][10] = po * t1 * t1 + ry * t4 * t4 + rz * t7 * t7;
m[5][5] = m[11][11] = po * t2 * t2 + ry * t5 * t5 + rz * t8 * t8;
m[6][6] = m[12][12] = 0;
m[4][5] = m[5][4] = m[10][11] = m[11][10] = po * t1 * t2 + ry * t4 * t5 + rz * t7 * t8;
m[4][6] = m[6][4] = m[10][12] = m[12][10] = 0;
m[5][6] = m[6][5] = m[11][12] = 0;
}

The difference are marked in bold.
As for the 2D system, only rotational inertia around Z axis is changed.

Mass matrix, so the influence of **consistent** mass inertia for nodes movement for different types of element are the following:

t = d*Ax*L; /* translational inertia */
ry = d*Iy; /* rotational inertia around Y axis */
rz = d*Iz; /* rotational inertia around Z axis */
po = d*J*L; /* polar inertia, assumes simple cross-section */

if ((n1z==1) && (n2z==1)) ////fixed-fixed
{
m[1][1] = m[7][7] = t / 3.;
m[2][2] = m[8][8] = 13. * t / 35. + 6. * rz / (5. * L);
m[3][3] = m[9][9] = 13. * t / 35. + 6. * ry / (5. * L);
m[4][4] = m[10][10] = po / 3.;
m[5][5] = m[11][11] = t * L * L / 105. + 2. * L * ry / 15.;
m[6][6] = m[12][12] = t * L * L / 105. + 2. * L * rz / 15.;
m[5][3] = m[3][5] = -11. * t * L / 210. - ry / 10.;
m[6][2] = m[2][6] = 11. * t * L / 210. + rz / 10.;
m[7][1] = m[1][7] = t / 6.;
m[8][6] = m[6][8] = 13. * t * L / 420. - rz / 10.;
m[9][5] = m[5][9] = -13. * t * L / 420. + ry / 10.;
m[10][4] = m[4][10] = po / 6.;
m[11][3] = m[3][11] = 13. * t * L / 420. - ry / 10.;
m[12][2] = m[2][12] = -13. * t * L / 420. + rz / 10.;
m[11][9] = m[9][11] = 11. * t * L / 210. + ry / 10.;
m[12][8] = m[8][12] = -11. * t * L / 210. - rz / 10.;
m[8][2] = m[2][8] = 9. * t / 70. - 6. * rz / (5. * L);
m[9][3] = m[3][9] = 9. * t / 70. - 6. * ry / (5. * L);
m[11][5] = m[5][11] = -L * L * t / 140. - ry * L / 30.;
m[12][6] = m[6][12] = -L * L * t / 140. - rz * L / 30.;
}
else if ((n1z==0) && (n2z==1)) ////pinned-fixed
{
m[1][1] = m[7][7] = t / 3.;
m[2][2] = m[8][8] = 13. * t / 35. + 6. * rz / (5. * L);
m[3][3] = m[9][9] = 13. * t / 35. + 6. * ry / (5. * L);
m[4][4] = m[10][10] = po / 3.;
m[5][5] = m[11][11] = t * L * L / 105. + 2. * L * ry / 15.;
m[6][6] = 0;
m[12][12] = t * L * L / 105. + 2. * L * rz / 15.;
m[5][3] = m[3][5] = -11. * t * L / 210. - ry / 10.;
m[6][2] = m[2][6] = 0;
m[7][1] = m[1][7] = t / 6.;
m[8][6] = m[6][8] = 0;
m[9][5] = m[5][9] = -13. * t * L / 420. + ry / 10.;
m[10][4] = m[4][10] = po / 6.;
m[11][3] = m[3][11] = 13. * t * L / 420. - ry / 10.;
m[12][2] = m[2][12] = -13. * t * L / 420. + rz / 10.;
m[11][9] = m[9][11] = 11. * t * L / 210. + ry / 10.;
m[12][8] = m[8][12] = -11. * t * L / 210. - rz / 10.;
m[8][2] = m[2][8] = 9. * t / 70. - 6. * rz / (5. * L);
m[9][3] = m[3][9] = 9. * t / 70. - 6. * ry / (5. * L);
m[11][5] = m[5][11] = -L * L * t / 140. - ry * L / 30.;
m[12][6] = m[6][12] = 0;
}

```
    else if  ((n1z==1) && (n2z==0))  ////fixed-pinned
    {
        m[1][1] = m[7][7] = t / 3.;
        m[2][2] = m[8][8] = 13. * t / 35. + 6. * rz / (5. * L);
        m[3][3] = m[9][9] = 13. * t / 35. + 6. * ry / (5. * L);
        m[4][4] = m[10][10] = po / 3.;
        m[5][5] = m[11][11] = t * L * L / 105. + 2. * L * ry / 15.;
        m[6][6] = t * L * L / 105. + 2. * L * rz / 15.;
        m[12][12] = 0;
        m[5][3] = m[3][5] = -11. * t * L / 210. - ry / 10.;
        m[6][2] = m[2][6] = 11. * t * L / 210. + rz / 10.;
        m[7][1] = m[1][7] = t / 6.;
        m[8][6] = m[6][8] = 13. * t * L / 420. - rz / 10.;
        m[9][5] = m[5][9] = -13. * t * L / 420. + ry / 10.;
        m[10][4] = m[4][10] = po / 6.;
        m[11][3] = m[3][11] = 13. * t * L / 420. - ry / 10.;
        m[12][2] = m[2][12] = 0;
        m[11][9] = m[9][11] = 11. * t * L / 210. + ry / 10.;
        m[12][8] = m[8][12] = 0;
        m[8][2] = m[2][8] = 9. * t / 70. - 6. * rz / (5. * L);
        m[9][3] = m[3][9] = 9. * t / 70. - 6. * ry / (5. * L);
        m[11][5] = m[5][11] = -L * L * t / 140. - ry * L / 30.;
        m[12][6] = m[6][12] = 0;
    }
    else ////Pinned-Pinned
    {
        m[1][1] = m[7][7] = t / 3.;
        m[2][2] = m[8][8] = 13. * t / 35. + 6. * rz / (5. * L);
        m[3][3] = m[9][9] = 13. * t / 35. + 6. * ry / (5. * L);
        m[4][4] = m[10][10] = po / 3.;
        m[5][5] = m[11][11] = t * L * L / 105. + 2. * L * ry / 15.;
        m[6][6] = m[12][12] = 0;
        m[5][3] = m[3][5] = -11. * t * L / 210. - ry / 10.;
        m[6][2] = m[2][6] = 0;
        m[7][1] = m[1][7] = t / 6.;
        m[8][6] = m[6][8] = 0;
        m[9][5] = m[5][9] = -13. * t * L / 420. + ry / 10.;
        m[10][4] = m[4][10] = po / 6.;
        m[11][3] = m[3][11] = 13. * t * L / 420. - ry / 10.;
        m[12][2] = m[2][12] = 0;
        m[11][9] = m[9][11] = 11. * t * L / 210. + ry / 10.;
        m[12][8] = m[8][12] = 0;
        m[8][2] = m[2][8] = 9. * t / 70. - 6. * rz / (5. * L);
        m[9][3] = m[3][9] = 9. * t / 70. - 6. * ry / (5. * L);
        m[11][5] = m[5][11] = -L * L * t / 140. - ry * L / 30.;
        m[12][6] = m[6][12] = 0;
    }

 /* rotatory inertia of extra beam mass is neglected */
for (i=1; i<=3; i++)     m[i][i] += 0.5*EMs;
for (i=7; i<=9; i++)     m[i][i] += 0.5*Ems;
```

The difference are marked in bold.
It looks quite complex, in reality we just eliminate the influence of mass inertia distributed along the element, for the rotation of pinned end node, as well as and movement along the local axis Y, so wherever index **2** and **6** for pinned node 1, or **8** and **12** for pinned node 2 appears.
If the reader notices any error in my reasoning, please correct me.

Pinned-pinned elements in dynamic analyses

For pinned-pinned elements is difficult to present deformation in different vibration modes, that’s why if dynamic analyses is performed (it’s optional in entire static & dynamic analyses) such element is divided for 2 parts: pinned-fixed and fixed-pinned, where the point of division lays in the middle of element length. In both “sub-elements” at least one rotating point exists, for sub-element 1 on second end, for sub-element 2 on first end, so based on those virtual node displacement in vibration modes the shape of Bézier curve can be generated for each element.
If dynamic analyses is skipped, pinned-pinned element stays as is, and is not divided for pinned-fixed and fixed-pinned halves, what makes static analyses results easier to interpret.
In both cases, in the AlfaCAD function presenting cross-sectional forces in an element, the element, regardless of any division that de facto creates a pair of elements, is treated as one continuous member.

Nodes radi

Node radii (if declared > 0) apply not only to rigid-rigid members, but also to pinned-rigid and pinned-rigid ones. However, their effect on a pinned-pinned members is ignored. In other words, the effective member length is reduced by the node radius only if the member on the node side is rigid.
By declaring a node radius and simultaneously declaring a pinned connection, it is impossible to determine the distance from the geometric intersection point of the elements at which the element's pin was designed. Therefore, it should be assumed that the node's pivot point lies at the geometric intersection point. This increases the designing element length despite the declared radius, so any calculation error is on the safe side. In design practice, if the node design differs, the resulting internal forces and deflections can be reduced.
However, the node radius affects the system's stiffness when the node also rigidly connects other system members.

Nodes restraint

Each node must be restrained in some way, this applies to each degree of freedom. Either a node is restraining by rigidly connecting the system's member, or it constitutes a rigid support, but none of the degrees of freedom can be released, otherwise, the system of equations cannot be solved. As long as the system constitutes a frame with rigid nodes, regardless of the support method, each node is stiffened by the system's elements. If the system constitutes a truss or a mixed system, meaning that for each member connected by a given node, the end of the member that meets in it is declared as a hinge, the node remains unrestrained for rotation. However, if this node is declared as a support node and the node's degree of freedom for rotation is taken away (rigid support), the system is solvable. A similar situation also occurs in a simple system when a single supported member is declared as a hinge at the support node, and the support is hinged. In this situation, the node is not rotationally restrained, and the system is not solvable.
Here is what Grok has to say:
Q: Does non restrained node for rotation due to connecting only pinned-pinned members, creates zero element on main diagonal of stiffness matrix, so system cannot be solved ?
A: Yes, in structural analysis using the stiffness method (e.g., for frames or finite element models), if a node has an unrestrained rotational degree of freedom (DOF) and is connected only by pinned-pinned members (which transmit no moments and thus provide no rotational stiffness), the corresponding diagonal element in the global stiffness matrix for that rotational DOF will be zero. This occurs because no elements contribute stiffness to resist rotation at that node.
A zero on the main diagonal (with the rest of the row/column also typically zero for that DOF) renders the stiffness matrix singular, meaning it cannot be inverted to solve the system of equations Ku=F for displacements u under loads F. This singularity indicates structural instability or a mechanism, as the structure can rotate freely at that node without resistance.

To resolve this:

- *Exclude rotational DOFs from the model if the structure behaves like a truss (only axial forces).*
- *Apply boundary conditions or restraints to the rotational DOF.*
- *Use moment releases or condensed stiffness matrices for pinned connections to avoid including unsupported DOFs.*

In practice, software often detects and handles such singularities by removing or condensing those DOFs, but in manual assembly, the system remains unsolvable as is.

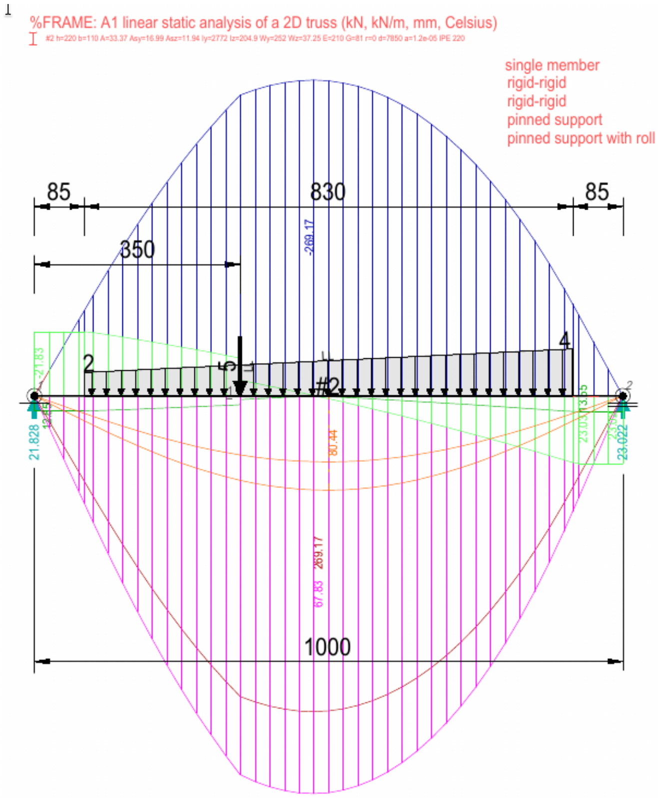
So like in the last sentence of the answer, AlfaCAD handles this situation by checking the restraint of all nodes, taking into account the type of support at the supported nodes. If the rotational degree of freedom of the node is not taken away by any of the elements connected at that node, and the support—in the case of supported nodes—does not take away this degree of freedom, the program automatically adds rotational restraint to the node. In this situation, removing a degree of freedom does not affect the calculation results (rotation of the node or its lack is not even noticed by element, due its hinged joint) but allows the creation of a stiffness matrix whose elements of the main diagonal are non-zero, thus enabling the solution of the system.

All is solved, time for tests

Below is an example, in many variants, of the simplest possible frame, which is actually a single-span beam, although in several tests, just for test purposes, it was alternatively treated as a composite beam consisting of two coaxial elements of identical length, rigidly connected. This solution allows for testing the system with pinned-rigid and rigid-pinned members, despite the simplicity of the system. For educational purposes, let's call it a frame, or a truss, which will be influenced by the beam's support method. As indicated by the previous considerations, if we are dealing with rigid nodes, we refer to it as a frame. If all nodes are pinned, we refer to it as a truss, although we know that in real engineering solutions, mixed systems are very often used.

A very simple system was chosen to facilitate the evaluation of calculation results. It should be remembered that the frame drawings show characteristic loads (in this case, only the live loads were applied, ignoring the self-weight of the frame), but the values of internal forces, support reactions, and stresses in steel cross-sections reflect the envelope of the design results for load combinations according to Eurocode at the ultimate limit state (ULS), while displacements and deformations reflect the envelope of the design results for load combinations at the serviceability limit state (SLS).
In Eurocode design, the live load is factored by 1.5 at the Ultimate Limit State (ULS), so such factor was used in all tests.

In all tests, the frame was subjected to identical loads: a trapezoidally distributed load along part of the member's length and a concentrated force load at a selected point on the member.



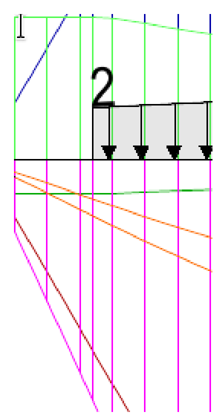
Frame A1 represents “classic” member with rigid ends, simply supported, it means pinned on the left, pinned with roll on the right.

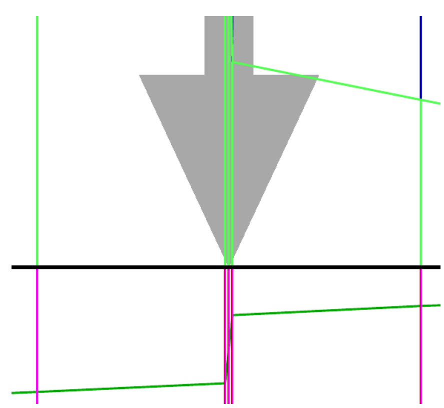
In static analyses document we can find:

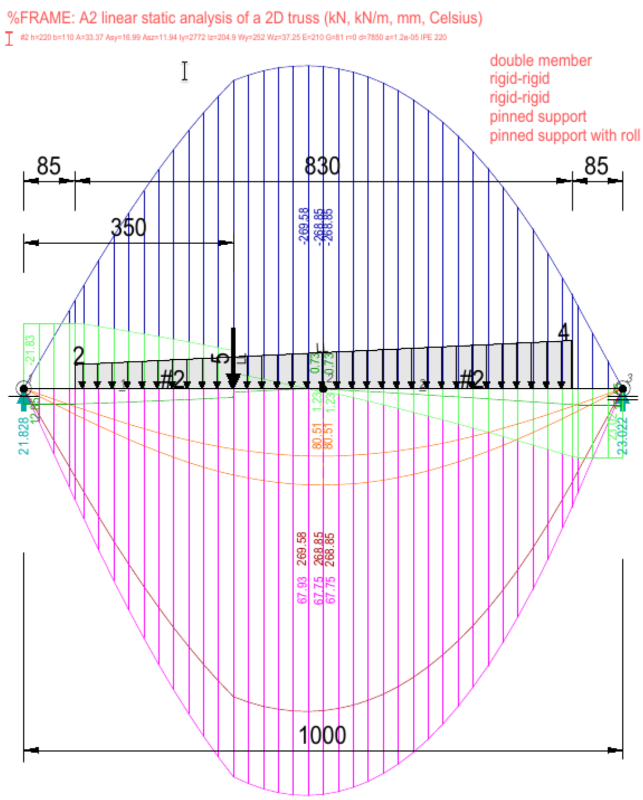
N O D A L   L O A D S + E Q U I V A L E N T   N O D A L   L O A D S (global)

Node	Fx	Fy	Fz	Mxx	Myy	Mzz
1	0.000	-14556.292	0.000	0.000	0.000	-29687453.858
2	0.000	-15343.708	0.000	0.000	0.000	29642871.142

There are Fy on both nodes but also Mzz, because member is rigid-rigid type, so initially the load the loading causes moments in the nodes, but after solving the system with boundary conditions such as supports, the nodes rotate freely, resulting in zero bending moments in the nodes.
It is worth noting that, due to trapezoidal integration, wherever the trapezoidally distributed load range begins or ends along the element's length, and such a point is not a point of division into design cross-sections with the dx discretization, such a point is added to the set of points at which the internal forces are determined.
In the case of a point force load, additional design cross-sections are added not only at the point of force application, but also at the virtually minimum distance before and after the point of force application (this distance is 5 mm or 0.2" in object units).
This is necessary for the correct presentation of the shear force envelope, as well as for the determination of cross-sectional moments and, consequently, stresses and deflections, with the smallest possible error resulting from trapezoidal integration.
Below an examples of accessories at the beginning of the trapezoidal application, the point of which is not supplied with the element for cross-sections with the value dx.
Another image shows design cross-sections for internal point force load.

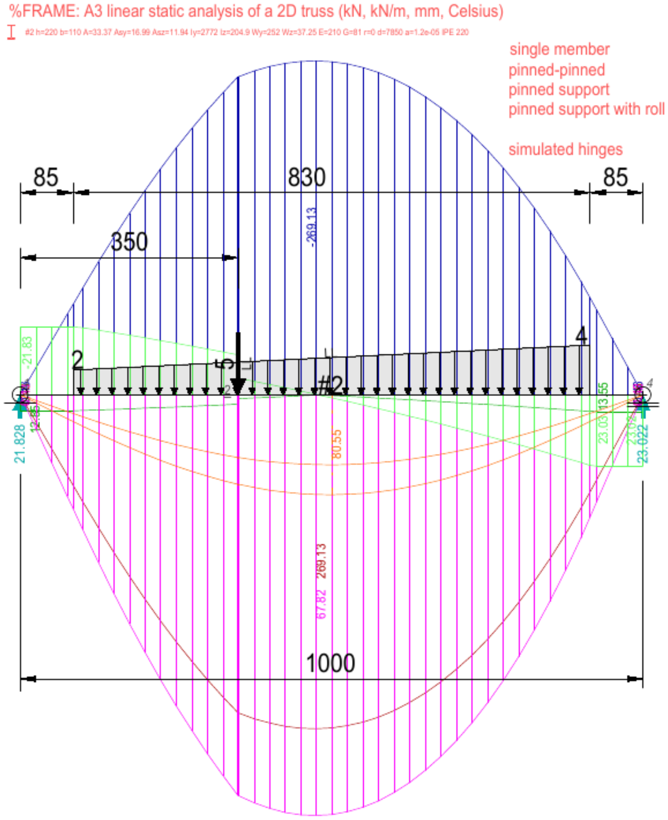


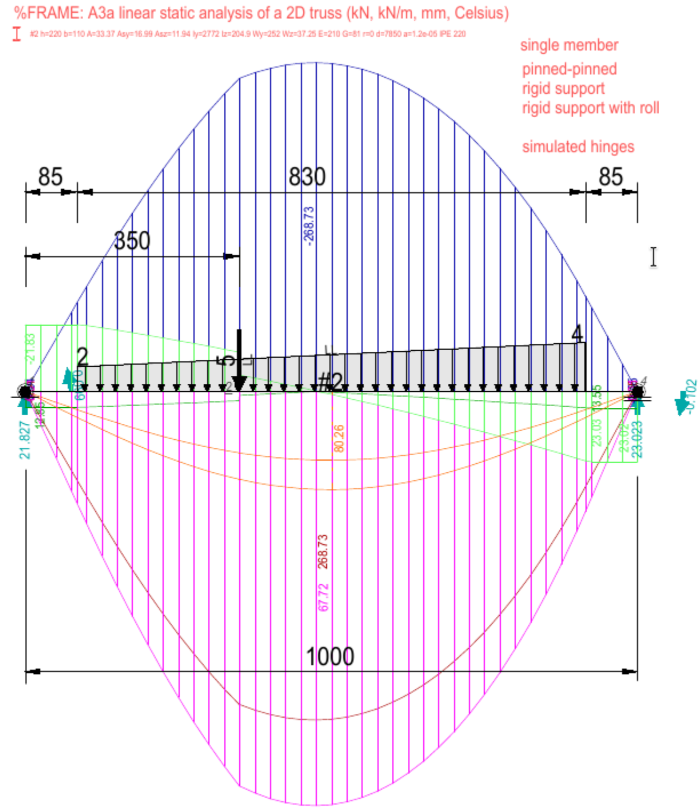


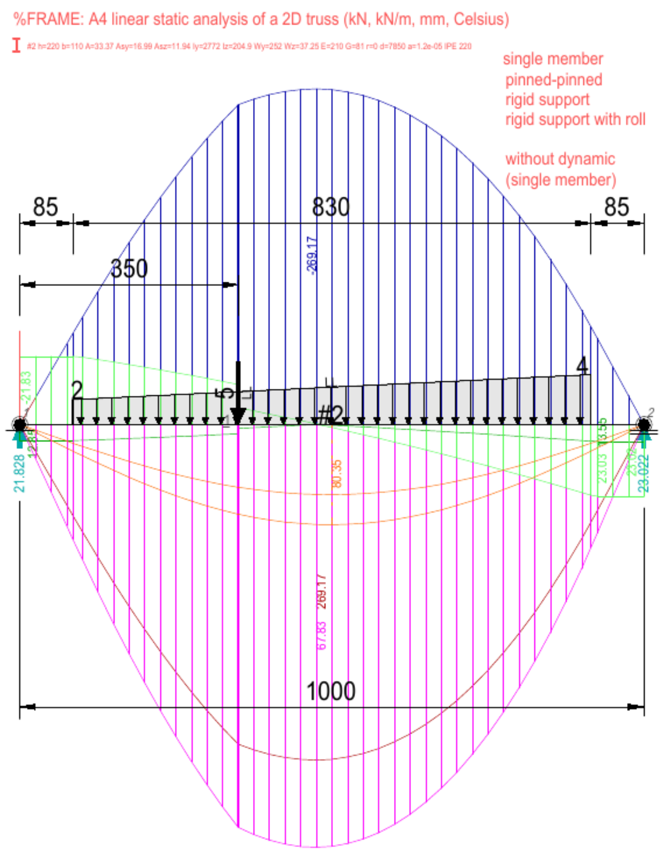


Frame A2 is the variant of A1 with this difference, that instead of single member, there are 2 members joined together and they both are rigid-rigid type. As is shown, the results (moments, shear forces, σ and τ stresses as well as deformations) are virtually identical, what was expected as for type of members introduced yet in the very first version of Frame3DD program.

Frame A3 and **Frame A3a** on the images on the left are variants of the Frame A1, with the difference that instead of a rigid-rigid member, a hinge-hinged element is used. The A3 variant uses pinned and pinned with roll supports, while A3a variant uses rigid and rigid with roll . In this case, so far no new type of elements were introduced, instead, both frame were solved using simulated hinges on the end of the member, so the solution is identical with the very first edition of Frame3DD.
As can be seen, the results (moments, shear forces, σ and τ stresses, and deformations) are virtually identical. As all members are rigid-rigid always, just very short extra members of very small stiffness for bending were created to simulate hinges, no matter what type of support is selected (fixed or pinned) every node is rotationally restraint anyway.





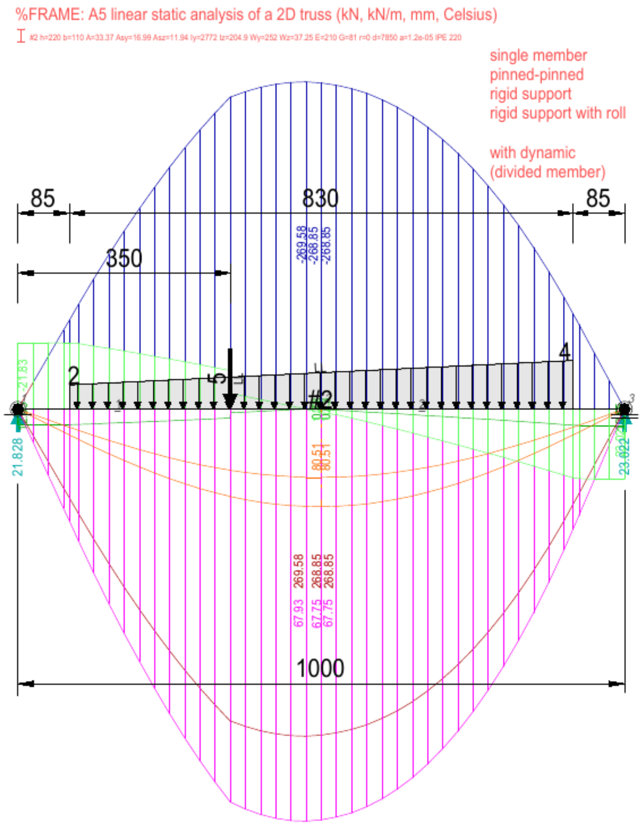


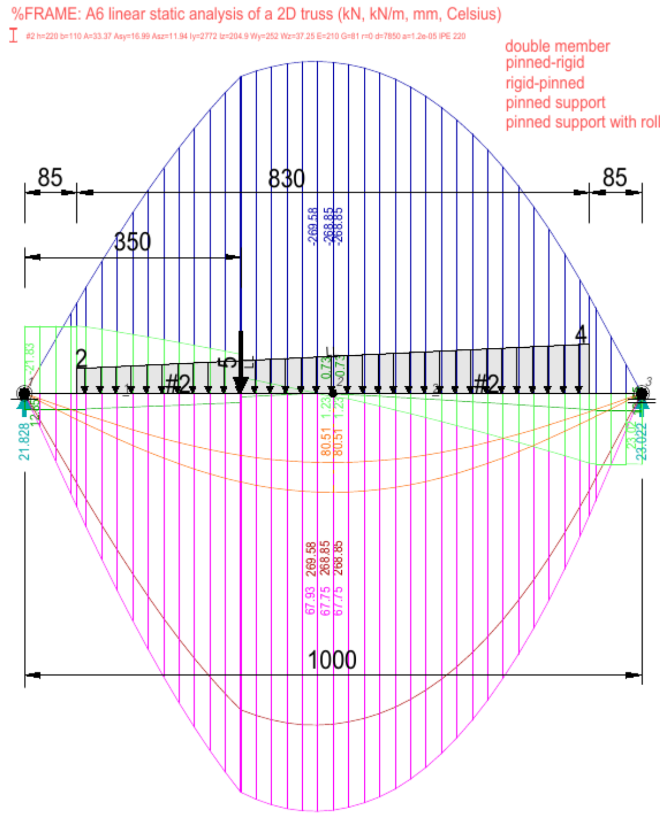
Frame A4 is a variant of A1 where new real hinged-hinged (or pinned-pinned) member is introduced.
As on the description (in red) on the image, this is single pinned-pinned member, with rigid and rigid with roll support.

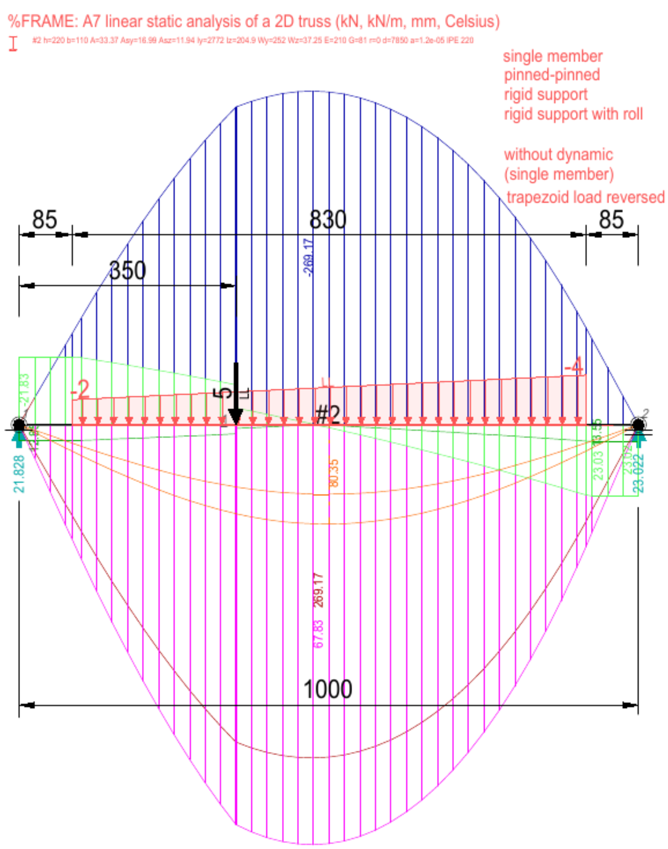
In case if pinned-pinned members, if only static analyses is performed without dynamic analyses, the type of member stays as is (pinned-pinned) what specially in case of classic trusses is very convenient way, due to minimal number of nodes and element.

In case of performing dynamic analyses together with the static one, due to dynamic analyses requirement for proper presentation (animation) of vibration modes, every pinned-pinned member is logically divided into two halves joined rigidly together, where one of them is pinned-rigid type and another is rigid-pinned.
This method increases number of nodes and number of members, but in cross section forces presentation function, such pair of element is treaten as a whole.

Frame A5 below is a variant of A1 (single members) but this time also dynamic analyses were performed, what led to division of single pinned-pinned member into 2 equal halves, creating pinned-rigid and rigid-pinned joined rigidly together. Rigid and rigid with roll supports were used. This system is in fact identical with **Frame A6** showed below, where intentionally used two members, pinned-rigid and rigid-pinned, where pinned ends lay on supports being rigid or rigid with roll types.
The results (moments, shear forces, σ and τ stresses, and deformations) are virtually identical with results in other variants, proving the validity of all the previously discussed assumptions regarding the stiffness matrix, load vectors, and methods for determining internal forces based on displacements obtained from solving the system of equations.



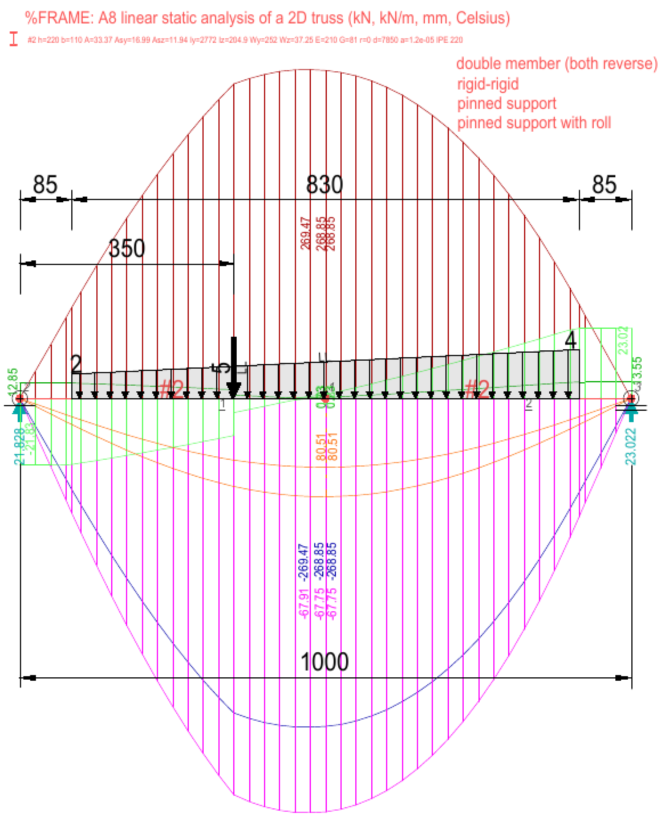


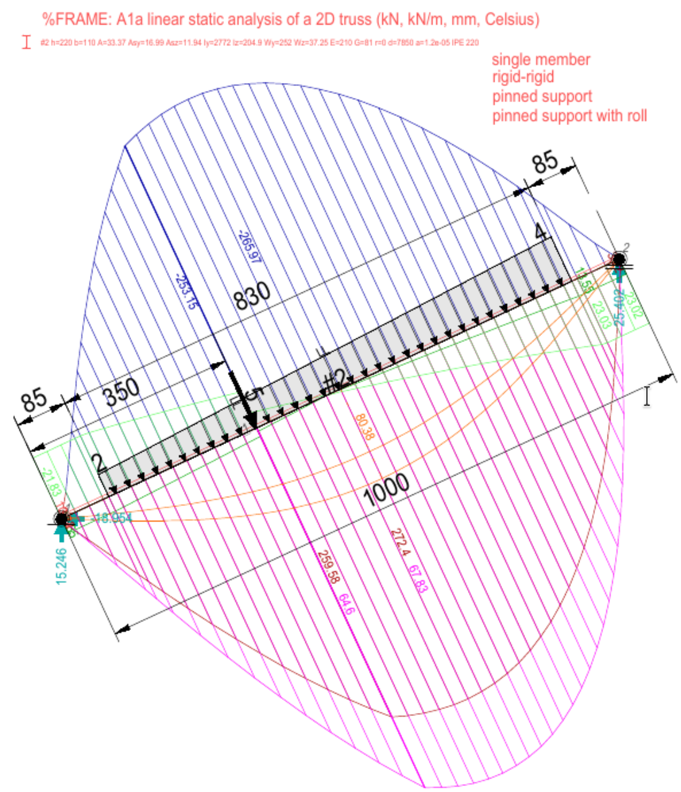


Frame A7 is a variant of A5 with this difference, that trapezoidally distributed load is inverted, it means the vector of load was drawn starting from right to the left, where member is drawn from left to the right. Load vector is marked in red. To maintain the correct sense of the force vector (pointing downward), the load values were set to negative. The arrowheads of the load vector show the actual sense of the forces.

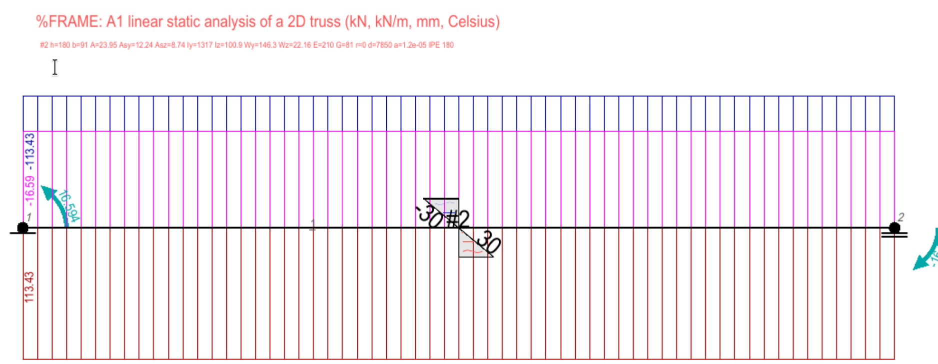
Frame A8 is a variant of A2 (two rigid-rigid members with pinned and pinned with roll supports) but this time both member vectors were reversed, so instead from left to right, they were drawn from right to left (marked in red), so opposite to trapezoidally distributed load vector.
Moments and deformations are shown always on the side where fibers are stretched, in this case, like in all cases – on the bottom side of the member, but shear forces and σ and τ stresses are shown in member's local coordinates, so if geometry of member is reversed, also graphs of shear forces and stresses are inverted.

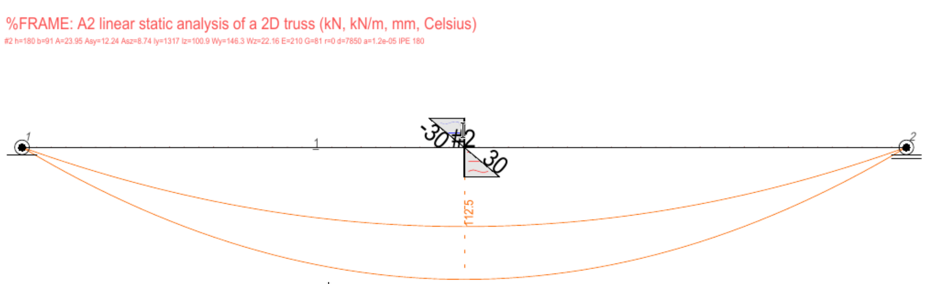
The final example is **Frame A1a** (below) as variant of A1, but rotated. The results are virtually identical, so the differences are negligible and result from a slightly different stiffness matrix due to the use of a horizontally sliding support, when—unlike the previous examples—there is a horizontal component of the loads in the system. A horizontal support reaction and an axial force in the element are present.

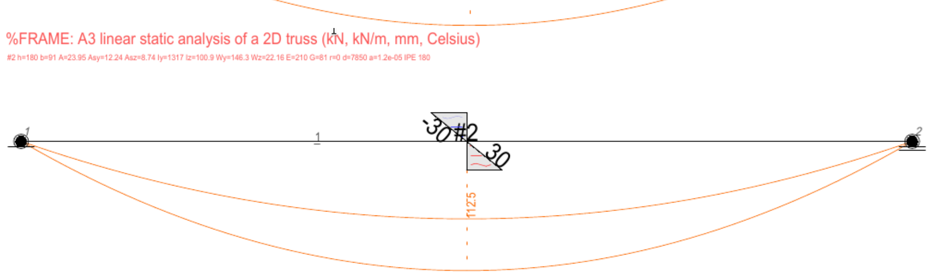




Thermal load





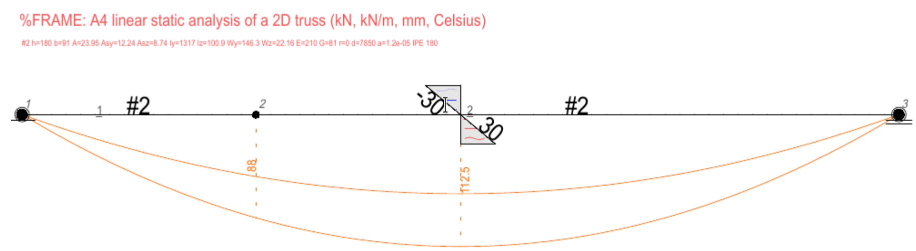


A1 is a beam (rigid-rigid member) fixed on both sides (fixed and fixed with roll supports) and subjected to a temperature gradient of -30°C on the upper side and +30°C on the lower side.

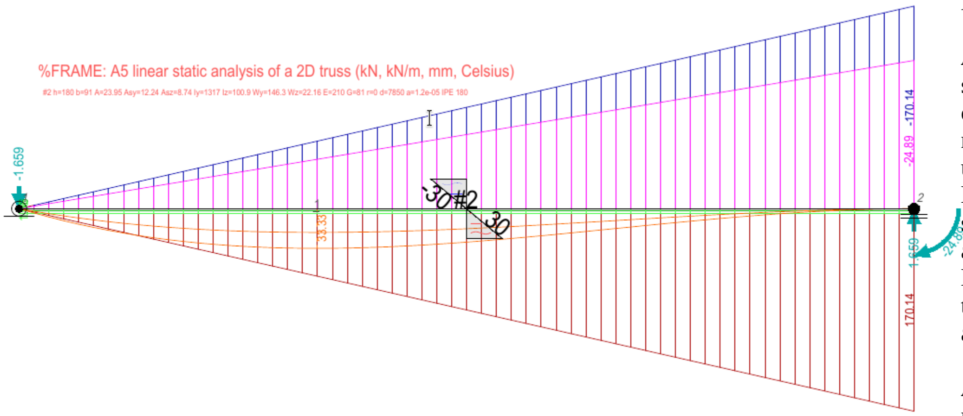
Due to the restraint, the beam cannot deform, and a bending moment develops in the beam. If such a distributed moment were applied to a simply supported beam, the beam would experience deformations of the same magnitude as the thermal deformations. Because the restrained beam experiences both a thermal load and a moment, the deformation effect cancels out and the beam remains undeformed, as confirmed by computational experiments.

A2 is the same beam (rigid-rigid member) but simply supported (pinned and pinned with roll).
Although the element is rigid-rigid, it is supported by nodes with freedom of rotation. Therefore, no bending moment develops within the beam, nor do shear or axial forces arise. Reactions also do not occur at the supports due to the lack of external loading. However, the beam does experience deformations.
The deformation magnitude corresponds to the deformations resulting from a distributed bending moment, such as that experienced in beam A1.

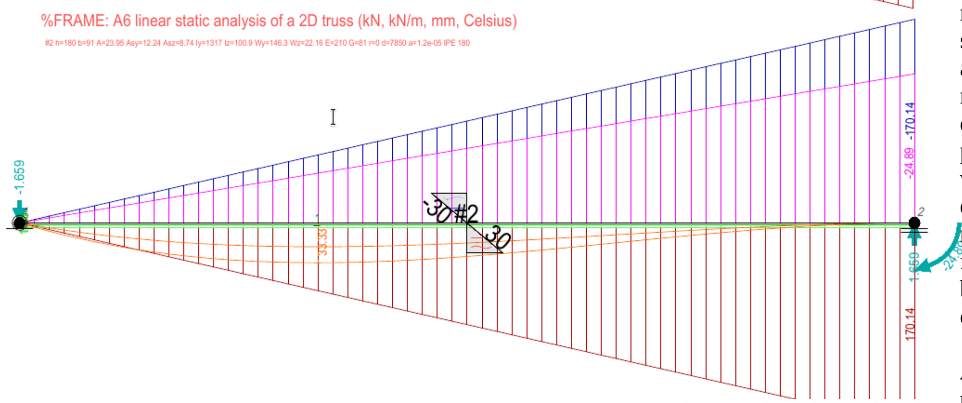
A3 beam is pinned-pinned member supported on rigid and rigid with roll supports. The static system is identical to the A2 beam.



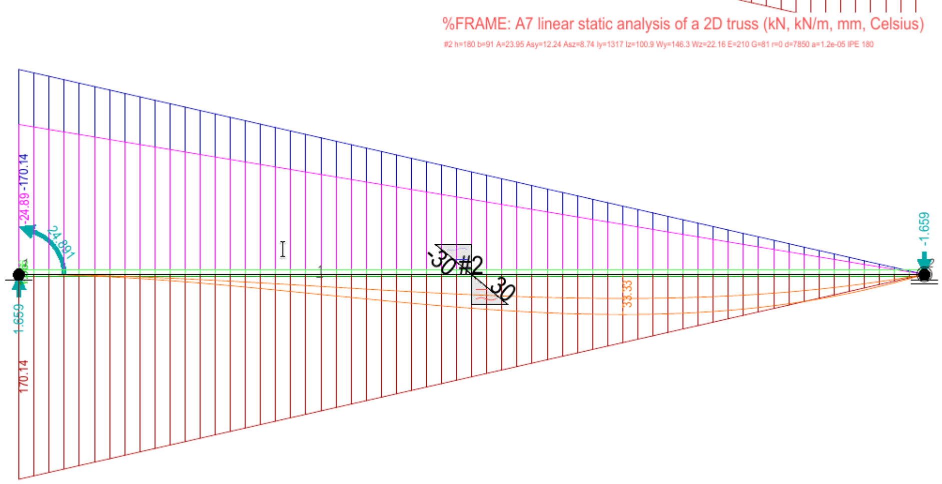
A4 A4 is a continuous beam constructed of two rigidly connected members of unequal length, on the left side there is a pinned-rigid member, on the right side rigid-pinned one, supported at opposite ends on rigid and rigid with roll supports.
Like the hinged-hinged element, the rigid-hinged and rigid-hinged elements, recently introduced as real types in Frame3DD, exhibit identical properties under thermal loading as a static system, as evidenced by identical beam deformations.



A5 is a beam as rigid-rigid element, simply supported on the left and fixed with a roll support on the right. Due to the rotational constraint of the node on the right, a bending moment is generated under the influence of internal thermal loads, which linearly approaches zero on the simply supported side of the beam. Although the element is declared as rigid-rigid type, there is no other element limiting the rotation of the support node, therefore the support node can freely rotate, eliminating the appearance of reaction moment.

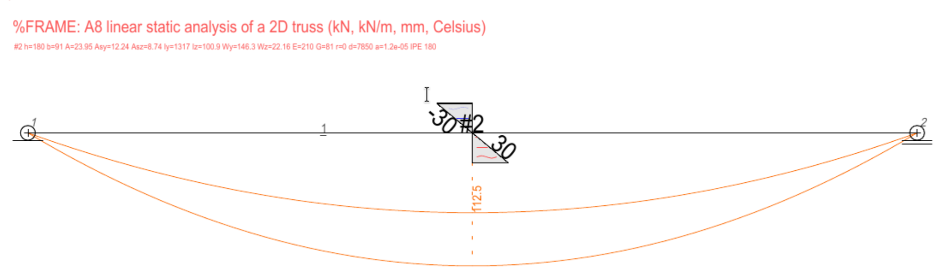


A6 is a variant of A5, with the difference that the member is declared as pinned-rigid, while the left support has changed its character and now serves as a rigid support. Due to the absence of other system members, as can be seen from the moment and deflection envelopes, a rigid connection with a pinned support is equivalent to a pinned element with a rigid support. However, the purpose of this demonstration is to present correct assumptions regarding the effects of internal forces resulting from thermal loading. The calculation results for beam A6 are identical to beam A5, despite the change in member and support type in the system.



A7 is inverted system, where the member is rigid-pinned, the left support is rigid, and the right is rigid with a sliding support. The system is inverted, but only partially, as the support types have not been swapped. Swapping the supports would not affect the distribution of forces and moments due to the absence of axial forces in the system. As can be seen from the moment and deflection envelopes, the system, as expected, experiences reversed moments and deformations relative to system A6, with the maximum values being identical.

In systems A5, A6 and A7, the response of the rigid support to the bending moment in the beam is the support moment, just as the response of both supports to the arising shear forces (green color of the envelope) are reaction forces of the same value, directed oppositely.

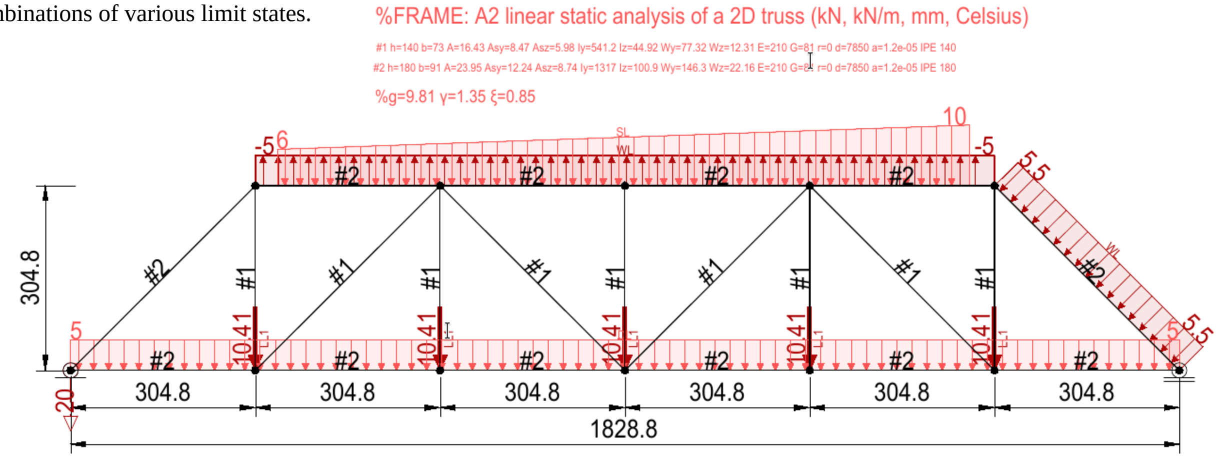


A8 is a variant of A3, with the difference that the member declared as pinned-pinned is now supported by a pinned and pinned with roll supports. The system was solved, and the results are visible in the deflection envelope, identical with results for A3, proving that AlfaCAD took care to restraint nodes, something neither the beam, being a pinned element, nor the supports, being pinned supports, did.

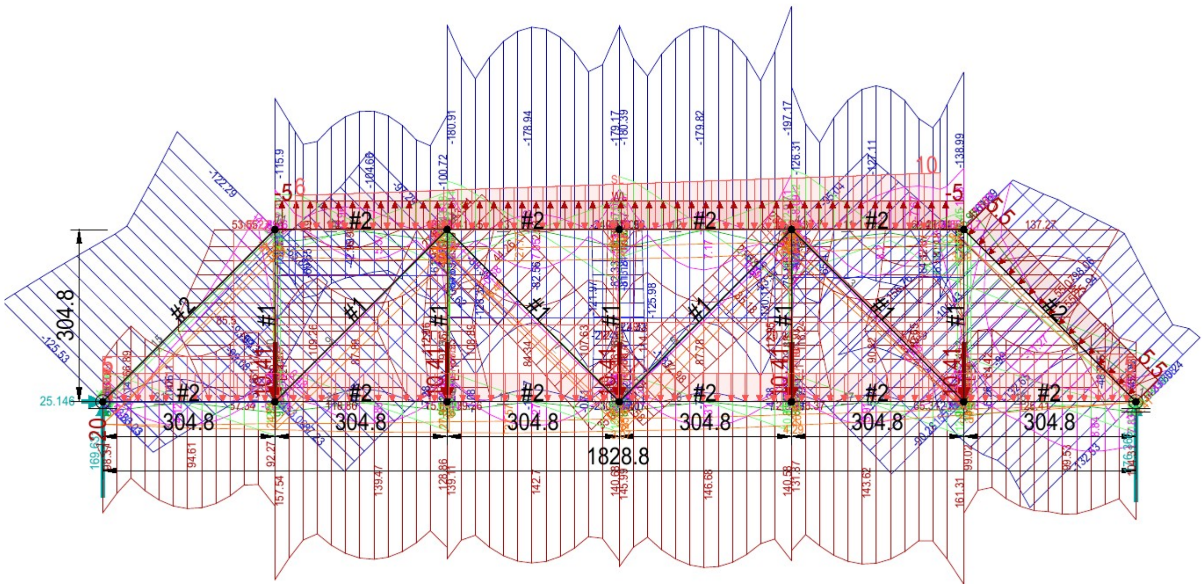
Frame, truss and semi-truss systems in practice

Truss as a frame

Below is a static diagram of a truss with rigid joints, effectively constituting a frame supported by a pinned support on the left and a pinned with roll support on the right. The frame is subjected to a set of loads, including self-weight, dead and live loads, snow and wind loads, and forced vertical displacement of the left support. These loads are test loads only and do not represent the loads of actual structures, but their diversity allows for the generation of envelopes of internal forces, stresses, joint displacements, and deformations in combinations of various limit states.

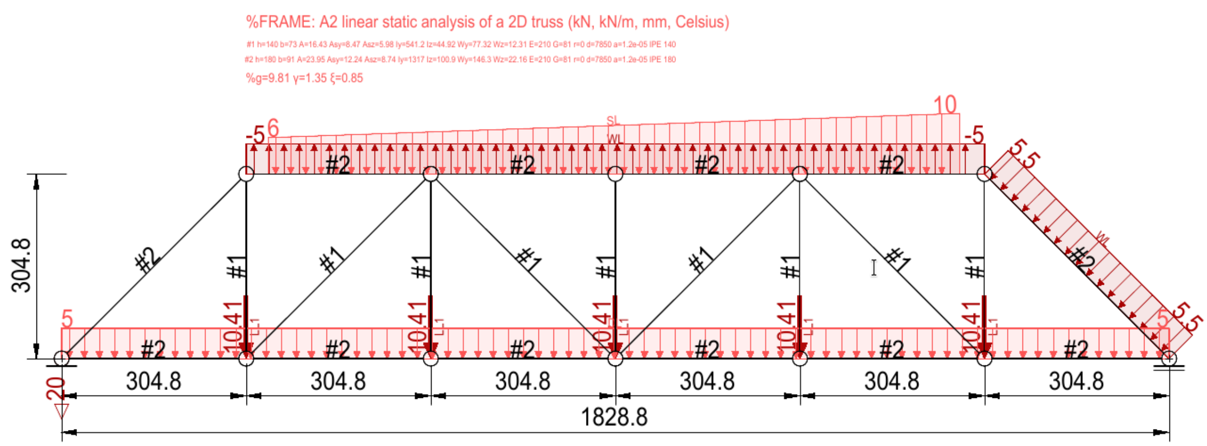


Below are the envelopes of internal forces, stresses in the ULS combination, and displacements in the SLS combination. The layers are not filtered, so all the envelopes are shown simultaneously, which does not make the drawing very clear, but our interest is basically only in the maximum stresses, which reach the maximum value of 197.17 MPa for compression and 161.31 MPa for tension.

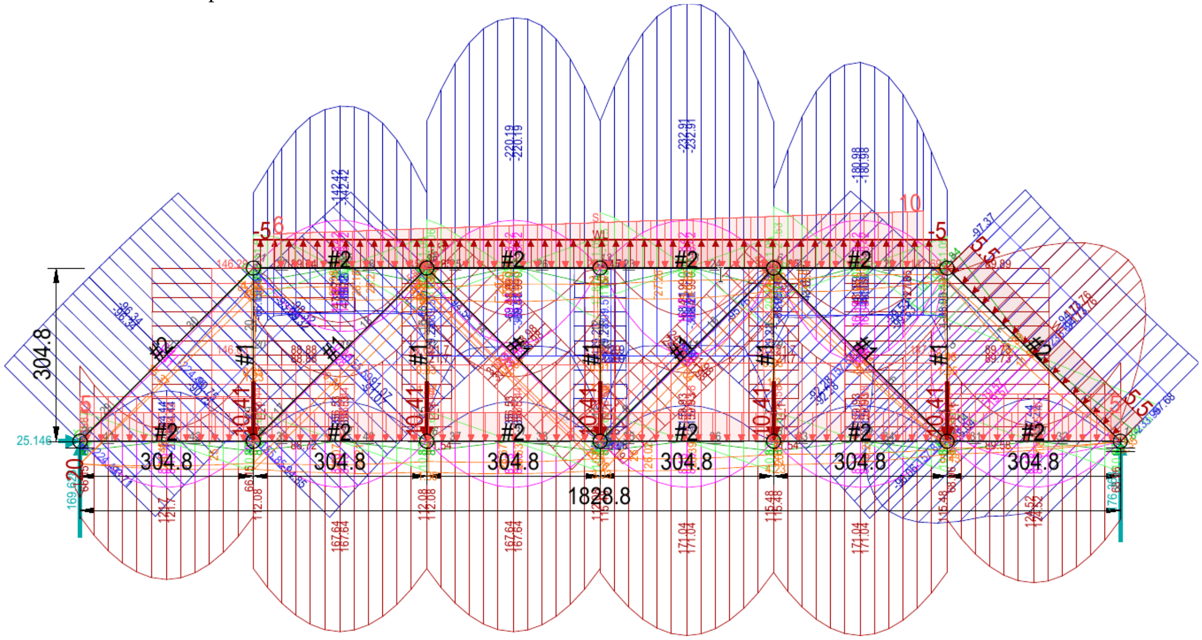


Truss as a real truss

Below is a static diagram of the same truss, but with all pinned joints and the same supports. The frame is subjected to an identical set of loads.

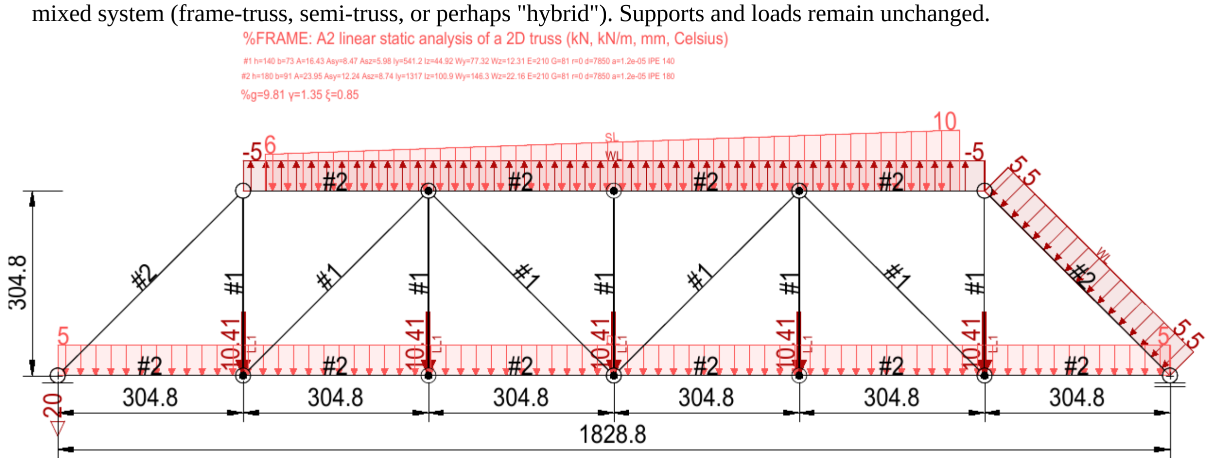


Below are the envelopes of internal forces, stresses in the ULS combination, and displacements in the SLS combination. The stresses reach a maximum value of 232.91 MPa in compression and 171.04 MPa in tension.
The stress increase represents 17.1% of the maximum stress values obtained in the frame variant.

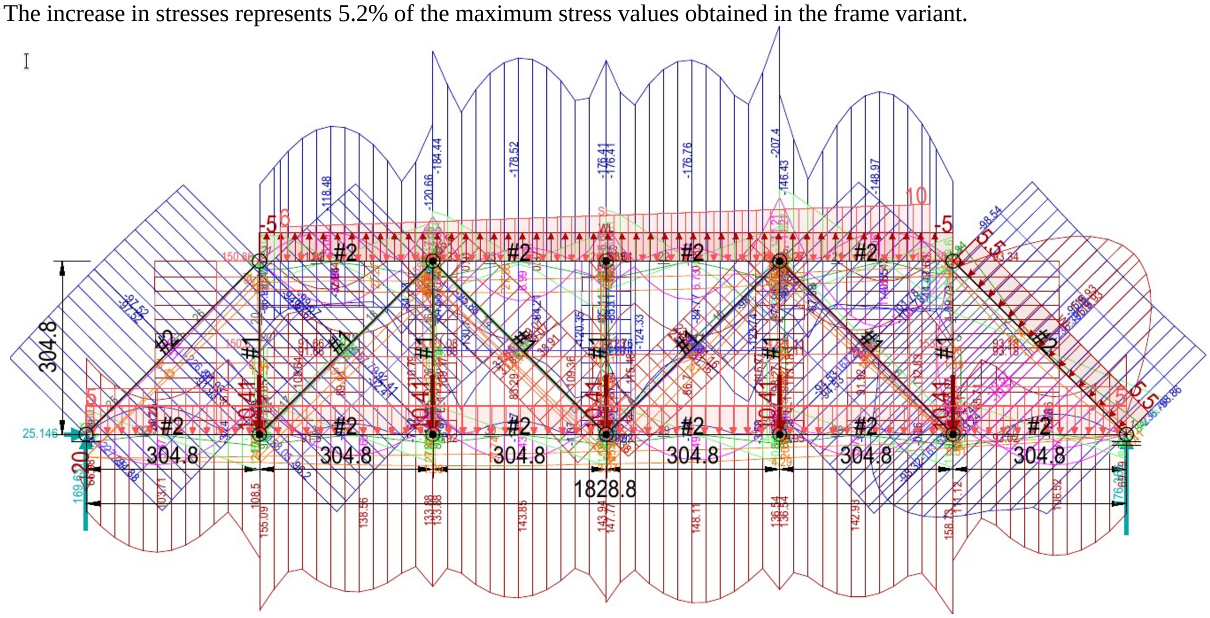


Truss as a semi-truss-semi frame

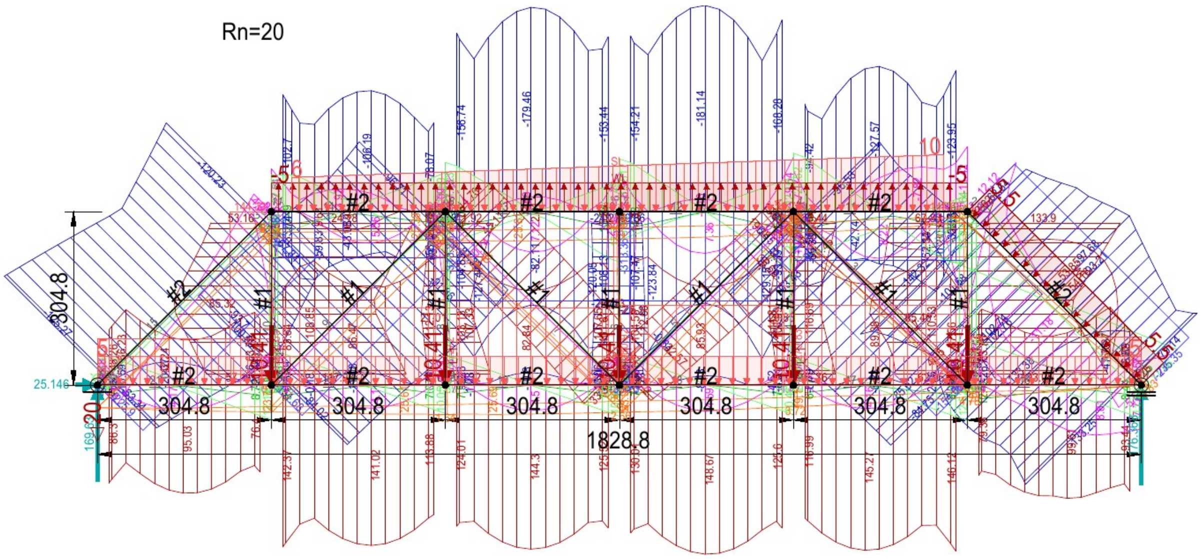
Below is a static diagram of the same truss, but with rigid top and bottom chords and other pinned-pinned elements, effectively creating a mixed system (frame-truss, semi-truss, or perhaps "hybrid"). Supports and loads remain unchanged.



The stresses reach a maximum value of 207.4 MPa in compression and 158.73 MPa in tension.
The increase in stresses represents 5.2% of the maximum stress values obtained in the frame variant.

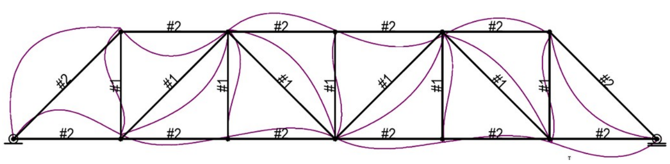


The ease of making changes to the static scheme and recalculating internal forces and stresses allows for variants, which not only allows for obtaining a structural model with properties closest to the actual structure, but also for obtaining an optimal solution or computational justification for the designer's chosen method of connecting system members.



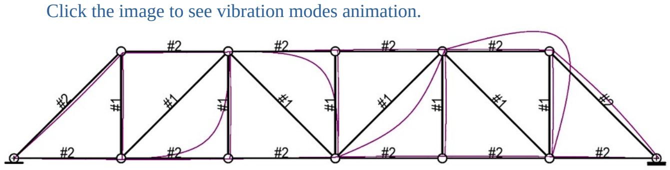
Below is the frame solution, i.e., the original system, with declared node radii. This assumes that each node in the system is formed by gusset plates with high bending stiffness in the frame plane, considered relatively high compared to the stiffness of the elements. Relatively high effectively means infinitely high, meaning that the node's deformability within the radius is negligible.
In this case, the maximum stress values have been reduced because in a system with rigid nodes with a given radius, the maximum stresses either occur within the element span or affect only cross-sections outside the node radius. In this variant, the maximum compressive stresses are 180.14 MPa and the tensile stresses are 148.67 MPa, which is an 8.1% reduction compared to the frame with zero radius nodes.

Dynamic analysis



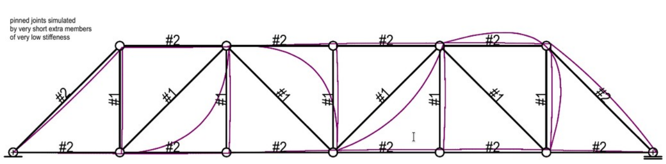
Click the image to see vibration modes animation.

A truss with rigid nodes (frame) under self-weight load, without taking into account additional masses from dead loads. Obtained mode frequencies: 20.67 Hz, 44.69 Hz, 59.20 Hz, 59.47 Hz, 62.38 Hz, 75.44 Hz, 76.51 Hz.



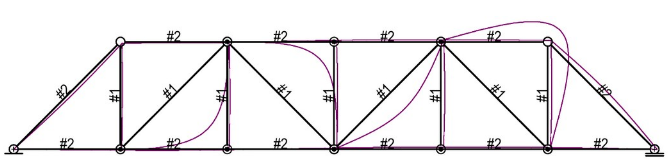
Click the image to see vibration modes animation.

A truss with hinged joints (all pinned-pinned members) under self-weight load, without taking into account additional masses from dead loads. Obtained mode frequencies: 19.93 Hz, 28.21 Hz, 28.71 Hz, 28.81 Hz, 30.69 Hz, 36.14 Hz, 37.20 Hz.



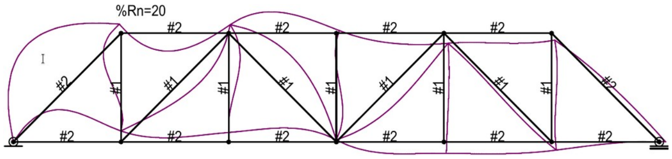
Click the image to see vibration modes animation.

A truss with pinned joints simulated with short, low-stiffness elements (all rigid-rigid elements, solution from the original Frame3DD version) under self-weight. Obtained mode frequencies: 17.55 Hz, 26.69 Hz, 27.52 Hz, 27.64 Hz, 29.64 Hz, 34.05 Hz, 35.52 Hz.
The vibration frequency values are very similar to the pure truss solution. This confirms the validity of the assumed stiffness and mass matrices for pinned-pinned members.



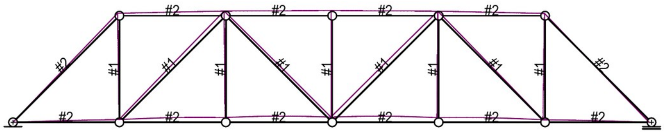
Click the image to see vibration modes animation.

A truss with rigid lower and upper chords and other semi-truss elements under self-weight. Obtained mode frequencies: 20.08 Hz, 28.21 Hz, 28.71 Hz, 28.81 Hz, 30.86 Hz, 36.18 Hz, and 37.23 Hz. The frequencies are almost identical to the pure truss solution, which is somewhat surprising. However, the mode shapes indicate that the lower and upper chords, stiffened by diagonals, are subject to vibration essentially as a whole, regardless of the chord segments connection method.



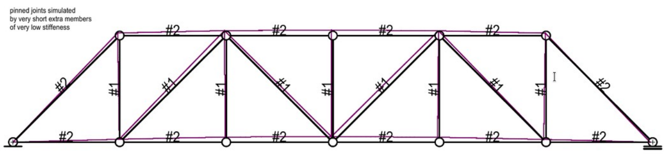
Click the image to see vibration modes animation.

A truss with rigid nodes (frame) with a given radius of all nodes of 20 cm, loaded with its own weight. Obtained natural frequencies: 22.06 Hz, 47.63 Hz, 62.96 Hz, 63.28 Hz, 66.05 Hz, 80.27 Hz, 80.86 Hz. The stiffening of the truss (frame) with nodes of infinite stiffness within the radius is clearly visible, which manifests itself in a significant increase in the natural frequencies.



Click the image to see vibration modes animation.

A truss with pinned joints (all pinned-pinned members) under self-weight load and masses from a dead load (DL). Obtained mode frequencies: 5.38 Hz, 10.67 Hz, 11.87 Hz, 12.05 Hz, 12.14 Hz, 12.17 Hz, 12.48 Hz. As the masses increase, the vibration frequency decreases significantly.



Click the image to see vibration modes animation.

A truss with pinned joints simulated with short, low-stiffness elements (all rigid-rigid elements, solution from the original Frame3DD version) under self-weight and masses from a dead load (DL). Obtained mode frequencies: 4.81 Hz, 10.68 Hz, 13.80 Hz, 21.40 Hz, 22.86 Hz, 24.21 Hz, 27.67 Hz.

The vibration frequency of the first three modes (the most important in dynamic analysis) is very similar to the solution for a pure truss, but the higher frequencies indicate a doubled harmonic basis. This means that the truss, due to some rotational damping at the nodes, does not tend to oscillate in various variants around 12 Hz, but around a doubled frequency of 24 Hz. This appears to be correct and not coincidental. This fact, and above all, the consistency of the lower fundamental frequencies, confirms the validity of the adopted stiffness and mass matrices for the pinned members.

Bottom line

This paper presents a methodology for analyzing and modifying the stiffness matrix of static systems, which, in addition to rigid-rigid members, also include rigid-hinged, hinged-rigid, and hinged-hinged (or pinned-pinned) members. The stiffness matrix in static systems, in the context of the Finite Element Method (FEM), represents the relationship between nodal forces and displacements in the analyzed structural system. This is a fundamental element in FEM, allowing for the calculation of the system's response to loads.

The load vector matrix, also known as the external force matrix, was also analyzed and modified, representing the forces and moments acting on the structure. This is a column matrix whose elements correspond to the values of forces and moments at individual points (nodes) of the system.

The next section modifies the trapezoidal integration method, which allows for the determination of forces and moments in element cross-sections based on the calculated values of nodal displacements and rotations, as well as the load on each element. Trapezoidal integration was also modified to determine the deformation of the element axis in each cross-section, and based on this, the deflections of the elements were determined.

A separate section was devoted to the analysis of thermal loading and its effect on internal forces and element deformations. A method for effectively calculating deflections due to the presence of internal forces in thermally loaded members was developed for members with a pinned node or nodes.

The analysis took into account the influence of rigid nodes with a given radius, and the influence of geometric stiffness and shear deformations was maintained throughout the entire system, regardless of the element type. The influence of shear deformations is currently omitted in static analysis performed in AlfaCAD, but will be included as an option in the next version of the program.

Substantial changes have been made to Frame3DD to extend the program to include pinned elements, eliminating the need for approximate calculations of such systems. In previous approach pinned nodes were replaced with short, low-stiffness members. This solution raises concerns due to the need to forgo consideration of shear deformations, geometric stiffness, introduces additional calculation errors, and requires the generation of additional nodes (one-third more in the case of pure trusses) and elements (three times more in the case of pure trusses), which results in poorer readability of calculation results. Despite this drawback, AlfaCAD has so far managed not only to automatically generate virtual members but also to hide them in the graphical presentation of calculation results. Now this drawback is over.

Of course, supplementing Frame3DD, AlfaCAD's static and dynamic analysis module, with pinned members, either unilaterally or bilaterally, is not a new solution and is quite common in structural analysis programs, but Frame3DD lacked this feature. It was unable to do so. It's doing it now.

Although the program is heavily modified, minor changes are necessary during data preparation, which AlfaCAD performs based on the system diagram of the system and system load diagrams.
In the text treated as the job name (the first line of the data file), it is recommended to specify the system of measurement, which is necessary to determine the optimal density of design cross-sections at the boundary of trapezoidal loads and at and around the point of application of concentrated loads. Generally, the keyword in the text is @UNITS= followed by the unit symbol, **SI** (International System of Units, also known as Metric System) or **IMP** (U.S. Customary System (USCS), also known as the Imperial System). If the keyword is missing, the SI system is assumed, and additional cross-sections are defined at a distance of 5 mm from the load range boundary or the point of force application. Example:
FRAME: A2 linear static analysis of a 2D truss (kN, kN/m, mm, Celsius) **@UNITS=SI**

The text preceding the keyword @UNITS= is arbitrary, although in AlfaCAD, the keyword FRAME: indicates the frame symbol following this keyword, ended with a whitespace, in this case "A2", and it serves as an indicator for defining the names of the drawing layers that will contain the envelopes of forces, moments, deflections, and stresses for various limit states. This is necessary due to the permissible multiple number of frame diagrams on a single drawing sheet. Therefore, frames must be distinguished from each other, and calculation results must be presented on different sets of layers.

The second change is to declare the node type of each element in the form of flags n1y, n1z, n2y, n2z meaning restraint the rotation around the local Y axis of node 1, restraint the rotation around Z axis of node 1, restraint the rotation around the Y axis of node 2, and restraint the rotation around the Z axis of node 2.
A value of 1 denotes restrained degree of freedom, while a value of 0 denotes non restrained degree of freedom.

An example of frame with rigid-rigid members:

```
# frame element data ...
21                              # number of frame elements
#e n1 n2 Ax    Asy     Asz      Jxx      Iyy      Izz      E      G     roll  density n1y n1z n2y n2z
#. . .  in^2  in^2    in^2    in^4    in^4    in^4    ksi   ksi  deg  kip/in^3/g . . . .
#. . .  mm^2   mm^2    mm^2    mm^4   mm^4    mm^4    MPa   MPa  deg  tonne/mm^3 . . . .

1 1 2 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
2 3 2 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 1 1 1
3 4 3 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 1 1 1
4 5 4 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 1 1 1
5 6 4 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 1 1 1
6 6 7 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 1 1 1
etc …..
```

An example of semi-truss with pinned-rigid, rigid-pinned and rigid-rigid members (truss with rigid bottom and top chords):

```
# frame element data ...
21                              # number of frame elements
#e n1 n2 Ax    Asy     Asz      Jxx      Iyy      Izz      E      G     roll  density n1y n1z n2y n2z
#. . .  in^2  in^2    in^2    in^4    in^4    in^4    ksi   ksi  deg  kip/in^3/g . . . .
#. . .  mm^2   mm^2    mm^2    mm^4   mm^4    mm^4    MPa   MPa  deg  tonne/mm^3 . . . .

1 1 2 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 0 1 0
2 3 2 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 0 1 0
3 4 3 1643 847 598 1 449200 5412000 210000 81000 0 7.85e-09 1 0 1 0
…
…
11 4 2 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 0
12 7 4 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
13 8 7 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
14 11 8 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 0 1 1
15 12 11 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 0 1 0
16 3 1 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 0
17 5 3 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
18 6 5 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
19 9 6 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
20 10 9 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 1 1 1
21 12 10 2395 1224 874 1 1009000 13170000 210000 81000 0 7.85e-09 1 0 1 1
```

The program generates output files similarly to the original version, but the range of abscissas (cross-sections) for which axial and shear force, bending moment and displacement (strain) values are reported, has been expanded and including points equal to the node radius (if declared), the beginning and end of a trapezoidal load, the point of application of a concentrated force, and cross-sections in its immediate vicinity before and after the application point.

The program also generates an additional output file exclusively for the generation of force envelopes in AlfaCAD.
The number of possible load cases has also been substantially expanded to meet the requirements of Eurocodes, ASCE, and ICC regarding load combinations in various limit states.

AlfaCAD provides an interface for the Frame3DDa module, which includes both data entry for calculations and the presentation of results. Therefore, knowledge of data file format is not necessary. The above information about changes in data file is intended for those adapting the module to work with another, optional interface.
The program's source code is available as the Frame3DDa.zip file at: https://github.com/MarekRatajczak2024/AlfaCAD

The last word

The author's work has not only a practical dimension, offering designers better computational tools, but above all it has an educational dimension, as there are few examples of open source software allowing for in-depth analysis of the numerical method of solving structural systems in both static and dynamic analysis. Most authors tend to protect their solutions.
Frame3DD started as a side-project when its author, Henri P. Gavin, today Ph.D., P.E., was a student of Hector Jensen in CE 512, Theory of Structures, at The University of Michigan. Entire program was developed at Duke's Pratt School of Engineering by Dr. Henri P. Gavin, with contributions from Dr. John Pye of the Australian National University.
The development of Frame3DD has benefited from many questions, observations, and suggestions from students and other users over a number of years.
Today I do not only benefit from their work. I continue it.

AlfaCAD, equipped with the Frame3DDa ("a" for "AlfaCAD") module, for static and dynamic analysis, has been implemented on five major computer platforms: Linux 64bit, Windows 64bit, Windows 32bit (in disuse), MacOS x64 and MacOS arm64, in 4 language versions: English, Polish, Ukrainian and Spanish. English, because approximately 19% of the world's population speaks English, either as a first or second language. Polish, because of author's mother tongue and because he obtained his diploma from the Wrocław University of Science and Technology in Poland. Spanish, because it’s one of the six official languages of the United Nations. Ukrainian, because if even one single house will be rebuilt after the war is over and AlfaCAD could contribute that, it was worth doing.

Enjoy AlfaCAD together with new look of Frame3DD.

Author

Marek Ratajczak

author@alfacad.net
https://alfacad.net

August 2025

Other articles on static analysis in AlfaCAD:

AlfaCAD 2.7 Supplement 4 (The World of the Steel equations)
AlfaCAD 2.7 Supplement 3 (The stirrup. An essence of simplicity)
AlfaCAD 2.7 Supplement 2 (A bit of concrete)
AlfaCAD 2.7 Supplement (Dynamic analysis)
AlfaCAD 2.7 (Static analysis, (almost) complete solution)
AlfaCAD 2.6 Supplement 2 (Static analysis extension)
AlfaCAD 2.6 Supplement (element properties)
AlfaCAD 2.6 (Back to the origin ...)

OK, this time just 99%

Links and sources:
https://svn.code.sourceforge.net/p/frame3dd/code/trunk/doc/Frame3DD-manual.html
https://people.duke.edu/~hpgavin/cee421/frame-element.pdf
https://www.uceb.eu/DATA/Books/Steel%20Designer%27s%20Manual,%206th%20Edition%20-%20(Malestrom).pdf
https://pdfcoffee.com/qdownload/steeldesignersmanualseventheditionpdf-pdf-free.html
https://www.engineeringtoolbox.com/statics-t_63.html
https://engineeringstatics.org/