Project G.A.M.E.

(New development of tower program)

Attachment

Specifications

Phase I

Specifications Phase I

„New development, Tower Program G.A.M.E. Version 1.0.0.1 “

Pos 1	**Phase I – New User Interface and Tower Geometry**
Pos 1.1	***New Design of UI*** (step iii): Implementation of templates and basic functionalities for a new graphical user interface (GUI) of the program GAME.
Pos 1.2	***Data Structure (Tower Geometry)***: Defining the detailed data structure for the handling of the static and design calculation as well as probabilistic analysis and DWG drafting. (For details see Annex, chapter “To Pos 1.2: …”)
Pos 1.3	***Exchanging graphic library*** (step ii): The graphic library HeidiDk10 (i386) of FMS has to be exchanged by another graphic library. This involves the implementation of new interface for input and output window graphics and, new definitions to all function and subroutine calls in all program modules to any graphic method. ***(a) Implement of basic methods***: Functionalities of view-methods: move, rotate, zoom, fixed viewpoints (front, top, etc.), select, highlight, hide, activate/deactivate, view modes (orthogonal, perspective, focusing), model view (modular, outline, rendering, shadow, lightning) ***(b) Enhancements to define the tower geometry***: Interactive definition of structure objects and object properties, synchronize in graphic views and tree or tabular views. Functionalities to display object properties, helps, information, remarks, attributes and/or comments in graphic.
Pos 1.4	***Exchanging of all embedded libraries*** (step i): All existing libraries in FMS, (e.g. ‘Units Converter.dll’; ‘MLColors.dll’; ‘Fonts.dll’; ‘Languages.dll’; ‘UserBucklingCurve.dll’),.have to be exchanged for libraries running on x64 platform.
Pos 1.5	***Databases of FMS*** (include as is, import, convert, eliminate): DBs: Conductor; Bolt; Panel; Profile; Bolt Fan; Profile Fan; Wind Drag; Wizard data file; RGL-file; Basic block. (For details see Annex, chapter “To Pos 1.5: …”).
Pos 1.6	***Redefining the editor modules***: Beside the mast editor module, the main program, there are four important editors for reading from and writing to databases: (1) Conductor DB; (2) Bolt DB; (3) Panel DB; (4) Profile DB. (For details see Annex, chapter “To Pos 1.6: …”).

ANNEX

Specifications Phase I

to Pos 1.2 / Pos 1.5 / Pos 1.6

Content

To Pos 1.2: Data Structure (Tower Geometry)

The data structure presented here is true for standard tower and standard tower with two tops:

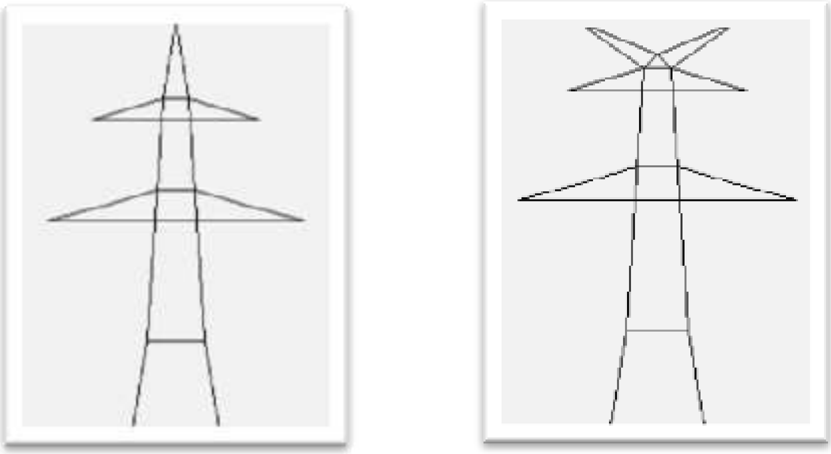


Fig. 1: a) “Standard tower” **b) “Standard tower with two tops”**

The main data groups of the data structure for these two tower types are:

(1) Tower Tops (Erdseilspitzen)

(2) Tower Body (Mastschaft)

(3) Cross Arms (Traversen)

In case of a tower family, that means a series of equal towers with different bottom tower body extensions, the data groups (1) Tower Tops and (3) Cross Arms and partially (2) Tower Body can be defined only ones and set as “Main Structure”. Therefore each variant is combined by this main structure definition and an individual (2) Tower Body extension definition.

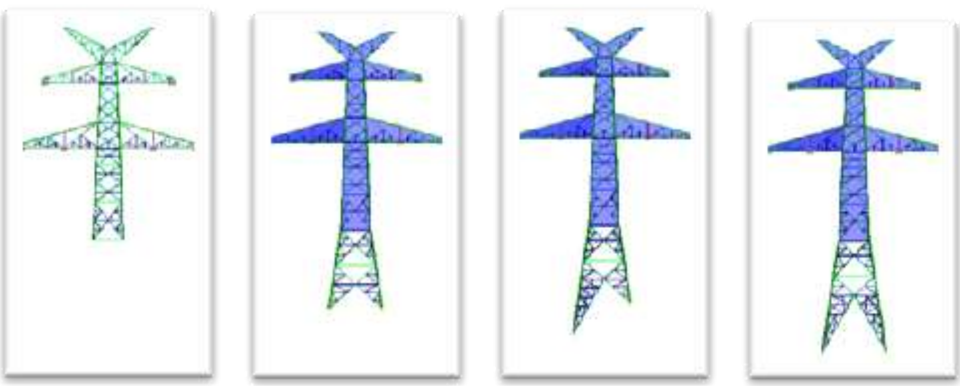


Fig. 2: a) Main Structure **b) “Variant 1”** **c) “Variant 2”** **d) “Variant 3”**

The so called leg extensions (Schrägfüße) are not a separate data group. It is only a special type of the bottom tower body section.

Comments

[DN] – (01.2015):

Illustrative Example

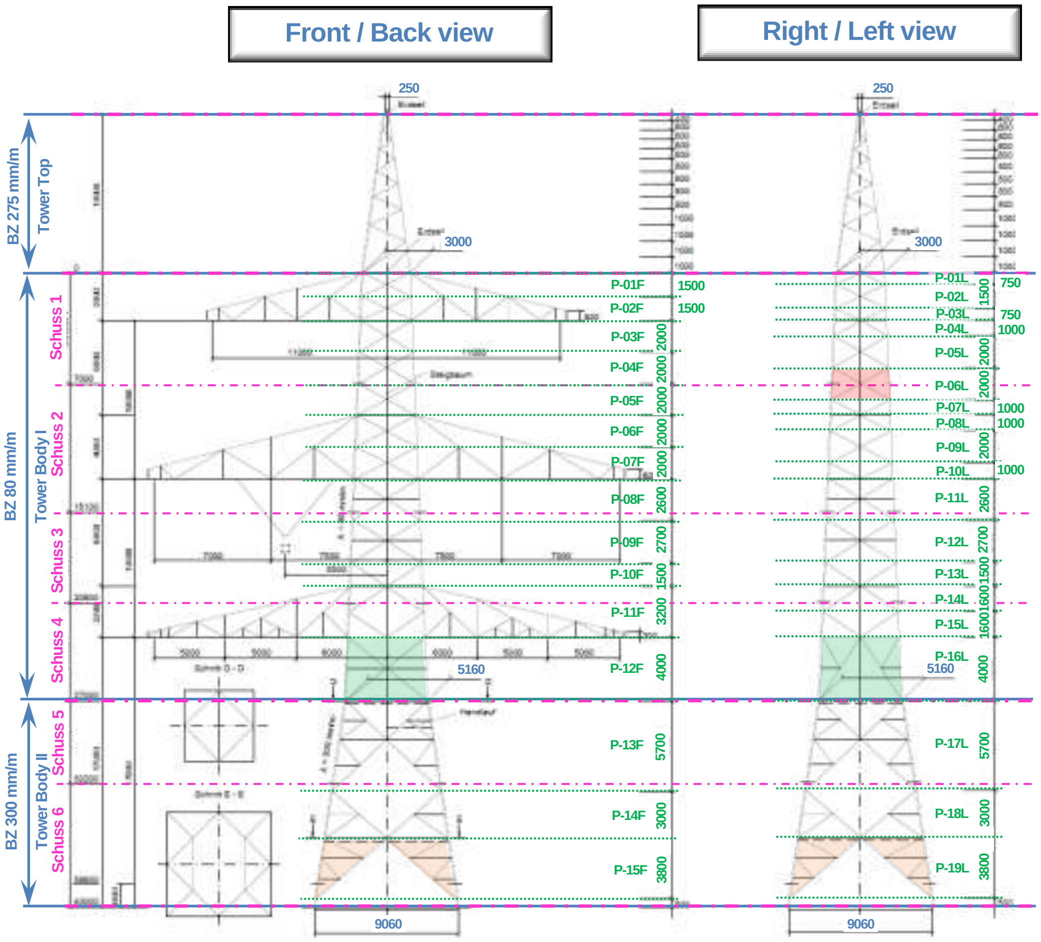


Fig. 3: Illustrative example: “Standard Tower“, with 3 BZ parts and 3 cross arms

BZ 1: Tower Top, (A1=250, A=3000)
BZ 2: (BZ 80 [mm/m]) Tower Body I, (A1=3000, A=5160), includes 4 sections (Schüsse);
BZ 3: (BZ 300 [mm/m]) Tower Body II, (A1=5160, A=9060), includes 2 sections, ‘Schuss 6’ takes the leg extension (Panel: P-15F and P-19L)

——— Width increase (BZ) level

-·-·-·- Section (Schuss) level

········ Panel (Feld) level

Remarks:

The bottom level of ‘**Schuss 1**’ is equal to the bottom level of panel **P-04F** and the top level of panel **P-05F** of the front view, but cut the panel **P-06L** nearly in the middle of the left view. The user has to define to which ‘Schuss’ these diagonals assigned to.

For ‘**Schuss 3**’, both levels don’t lie on panel levels, but the assignment of the panels is simple, e.g. left view, **P-11L** assigns to ‘Schuss 2’ and **P-13L** assign to ‘Schuss 3’.

The panel names are only as an example, but in general numbered from top to bottom.

Comments

[DN] – (01.2015):

Data Structure: (1) Tower Tops (Erdseilspitzen)

Tower Top for "Standard Tower"

In case of "standard tower", the tower top is in general only one tower section (Schuss) of equal BZ (width increase), with mirrored front and back part and mirrored left and right part.

The corner members at each 4 edges have the same properties. The properties of the panel members are in general equal for the front and back view as well as for the left and right view.

Exceptions:

(Sometimes) The front and back or the left and right parts are not mirrored nor have equal panel members.

(Rarely) The tip of the tower top has a special configuration to take place for conductors. This implicates more than one section. (As an example, see Fig. 4, "Erdseillaterne".)

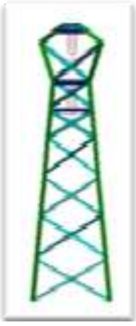


Fig. 4: Erdseillaterne

Tower Top for "Standard tower with two tops"

In general the top on one side is a copy (and rotated by 180 degrees) of the top from the opposite side, likewise as a cross arm. The bottom areas of the tops are connected at the side view areas of a symmetric pyramid with a top width length of A1=0 (see Fig. 5).

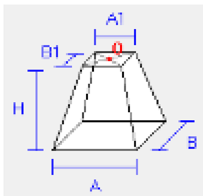


Fig. 5: Sym. Pyramid

The corner members of the inner and outer edges have often different properties. Therefore two corner member types exist, the inner corner members and the outer corner members.

Exceptions:

(Rarely) There exists more than one section for each tower top.

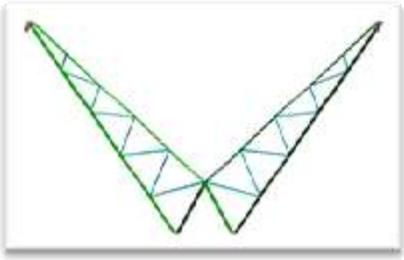


Fig. 6: Two tops example

(Occasionally) The tower tops are different.

(Occasionally) The tower top block type has a triangular cross section and therefore has only three views (front / back / side) not four (front / back / left / right).

Comments

[DN] – (01.2015):

Data Structure: (2) Tower Body (Mastschaft)

General

The data structure for “Tower Body” includes the standard tower body sections (Mastschaft Schüsse) as well as leg extension (Schrägfuß). The standard tower has equal leg lengths and symmetrically leg configuration.

Different width increases (BZ) divide the tower body in clear cut volume sections. All member types are clear positioned in such BZ-section, without any exception. Therefore the data structure tree view includes BZ titles as next substructure after tower body.

A section (Schuss) defines the distance between two coupling joints of corner members.

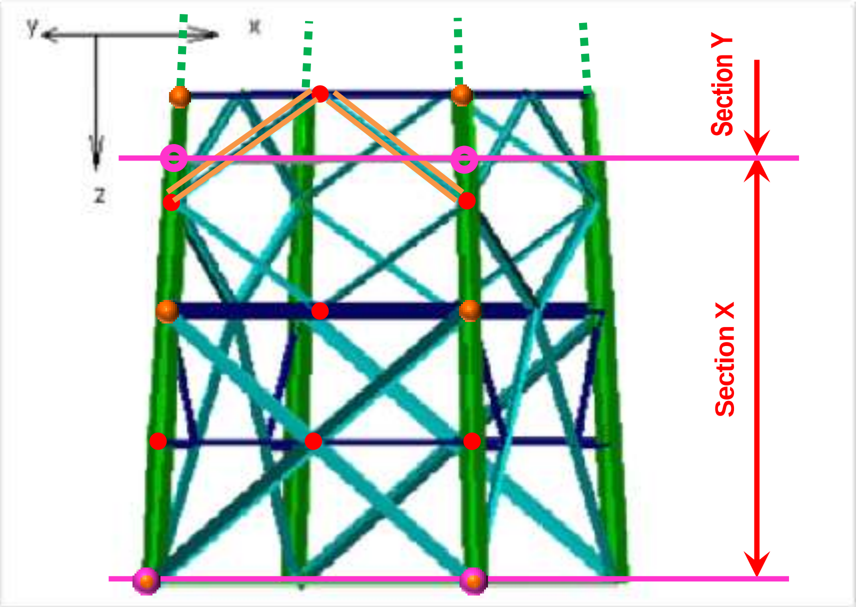


Fig. 7: Tower body part with marked nodes at the front view

The coupling joints within a BZ in general are not placed at an edge joint of a panel or a coupling joint of another member type, because they are often very complex constructions (see Fig. 8). But all edge points of a BZ are coupling joints of corner members.

Therefore, at the cut level of a section, all other member types as diagonal members, redundant members and brace members, they can be assigned often to more than one section. Here an individual choice is needed, e.g. the marked diagonals in Fig. 7 are assigned to section Y.

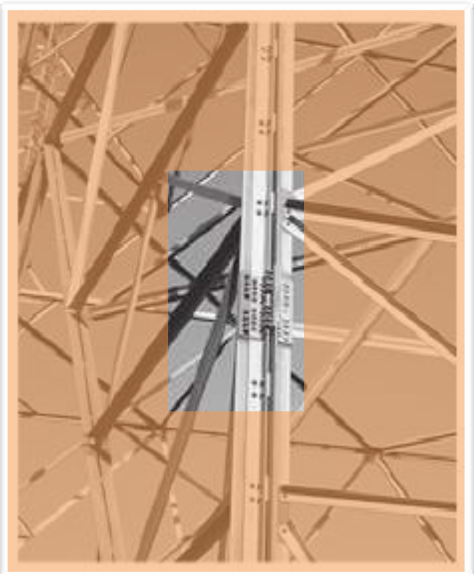


Fig. 8: Coupling joint of corner member

Comments

[DN] – (01.2015):

Comments

[DN] – (01.2015):

Tower Body “Standard Tower Body”

Width increase (Breitenzunahme, Breitenzunahmen *pl*)

The standard tower body has $i=1$ to $i=N$ width increases.
(BZ := Breitenzunahme).

Section (Schuss, Schüsse *pl*)

In general each BZ is quadratic and has $j=1$ to $j=S$ sections. All four corner members of a section have the same properties.

The cut level of a section is always horizontal, respectively all corner coupling nodes of one section to another section are at the same altitude.

Panel (Feld, Felder *pl*)

Within a section all four corner members have equal properties and enclose $k=1$ to $k=P$ panels.

A section has commonly symmetric or mirrored panel configuration of front and back view as well as of right and left view.

Exceptions:

(Rarely) The panels of the front/back view can be partially shifted at the half panel height to the panels of the right/left view.

(Sometimes) The front and back or the left and right parts are not mirrored nor have equal panel members.

(Occasionally) Not all corner elements have the same profile.

Illustrativ example Fig. 3

BZ	Section	Panel (Front/Back)	Panel (Left/Right)
BZ 80 mm/m Tower Body I	Schuss 1	P-01F to P04F	P-01L to P-06L
	Schuss 2	P-05F to P08F	P-07L to P-11L
	Schuss 3	P-09F to P10F	P-12L to P-14L
	Schuss 4	P-11F to P12F	P-15L to P-16L
BZ 300 mm/m Tower Body II	Schuss 5	P-13F	P-17L
	Schuss 6	P-14F to P15F	P-18L to P-19L

As suggestion for the disposition of tower body in Fig. 3:

BZ: The tower body in Fig. 3 has $N=2$ width increases.

Schüsse: Tower Body I (BZ 1) $S=4$ (‘Schuss 1’ to ‘Schuss 4’),

Tower Body II (BZ 2) $S=2$ (‘Schuss 5’ and ‘Schuss 6’).

Felder: The front/back panels and the right/left panels of ‘Schuss 1’ and ‘Schuss 2’ up till the last one, (P-08F of the front view and P-11L of the left view), are shifted.

The panels P-08F to P-11F and P-13F to P-15F from the front view have the same configurations as the panels P-11L to P-15L and P-17L to P-19L from the left view.

Only the front view panel P-12F and the left view panel P-16L has different layouts, but are at the same levels, (see marked green areas in Fig. 3).

Even if the panel of the front/back view is equal to the right/left panel configuration, the diagonal elements or redundant elements or brace elements of the front/back panel have very often different profiles to the right/left panel. E.g. the diagonal elements of F-09F have the profile L200x18 and the diagonal elements of P-12L have the profile L180x16.

Bottom Tower Body "Leg Extension"

Each tower body has a "Leg Extension" part. This part is the tower body part below the undermost horizontal bracing. In Fig. 3 it is the marked orange areas.

The leg extension part consists of 4 triangular pyramid volume blocks, one for each edge with the tip at the support node. Often these 4 blocks have equal geometrical properties, (widths and height), as well as the triangular panel at each view. Rarely these 4 triangular pyramid blocks are of different heights and therefore of different panel configuration (see Fig. 9).

The top couple nodes of all 4 corner members are at the same height level, but in general with a distance to the panel edges.

The diagonal members of a leg extension edge body go always from the support node to the undermost horizontal bracing (see Fig. 10), even if the connection node at this horizontal bracing looks like a crossing node (see Fig. 9).

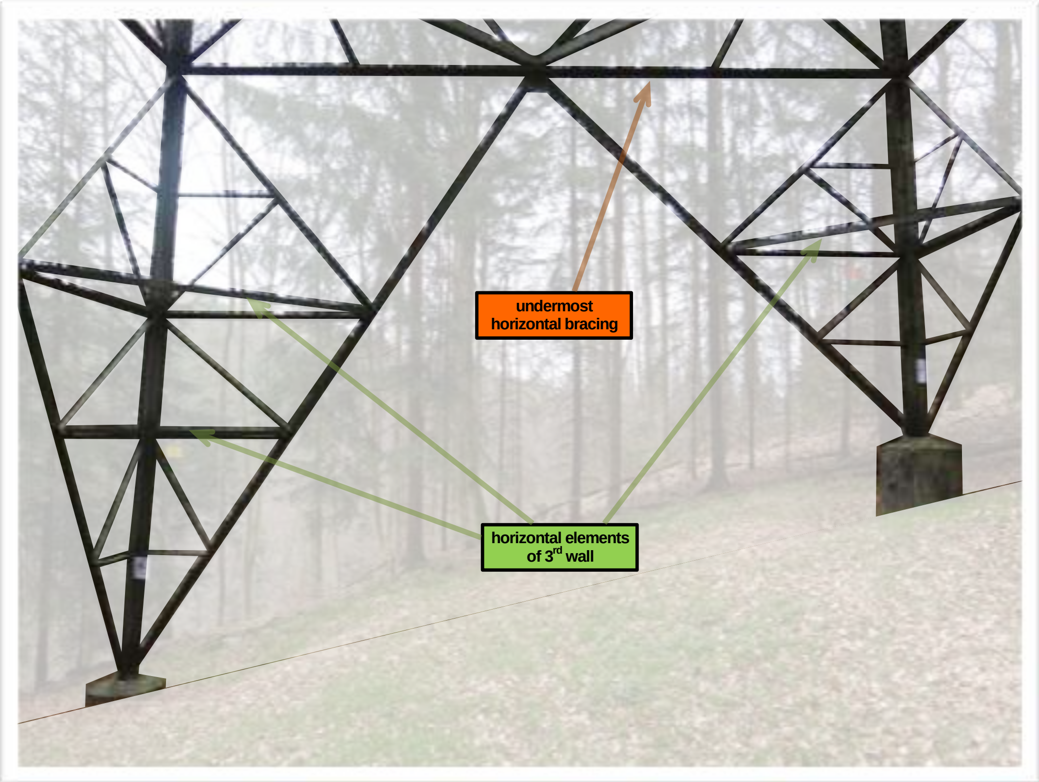


Fig. 9: Right side of a leg extension of a real existing mast (as an example)

Comments

[DN] – (01.2015):

Exceptions:

(Often): One or all of the 4 triangular leg extension pyramid bodies have element members at the inside body face, the so called 3rd wall (e.g. see Fig. 9 one horizontal element at the 3rd wall of each extension exist; in Fig. 10 there are two horizontal elements inside at the normal leg and three at the positive leg extension, at the negative leg extension are only diagonal members outside).

(Occasionally): The diagonal members from the left and right triangular panel of a view face are not connected in one node at the undermost horizontal bracing, they are connected with a gap, which is centered to the tower width.

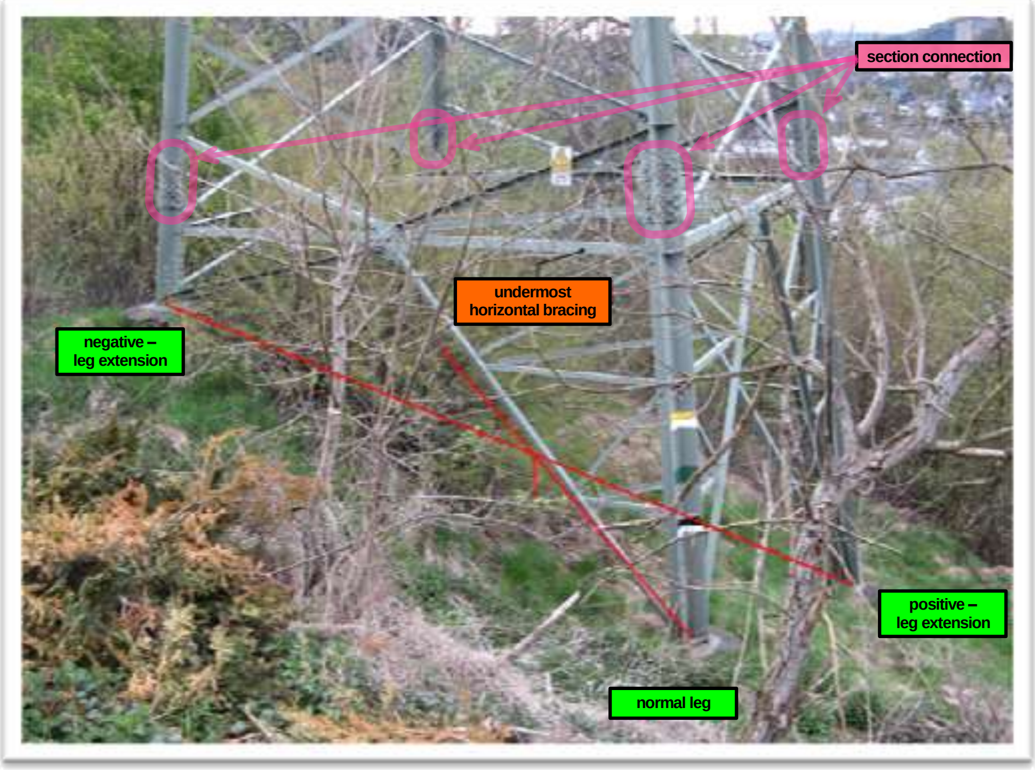


Fig. 10: Tower body leg extensions on a slope

Comments

[DN] – (01.2015):

Data Structure: (3) Cross Arm (Traverse)

In general a cross arm consist of a left and a right half with their center axis parallel to the global X-axis (see Fig. 3 “Front/Back view”) and a horizontal lower girder (see Fig. 11). The halves are identical; each of them is a copy and rotated by 180° about the vertical tower body axis of the other.

The front and back views of a cross arm have identical panels.

Depending on the cross arm length one or more section borders exist. The horizontal distance of the section border from the top girders to the center axis of the tower body differ to the horizontal distance of the section border from the bottom girders (see Fig. 11).

Conditionally on design, the corner points of the panels from the bottom or top girder in general differ to the corner points of the panels from the vertical view up to the corners of the cross arm body.

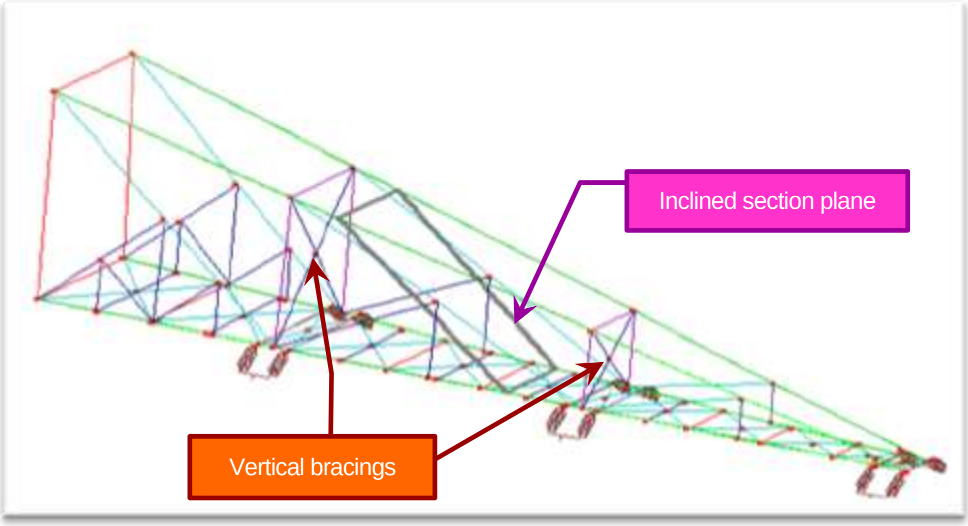


Fig. 11: Typical cross arm construction, here with three conductor attachment points, two vertical bracings and two sections (Schüsse) for the upper and lower corner elements at different distances to the center of tower body.

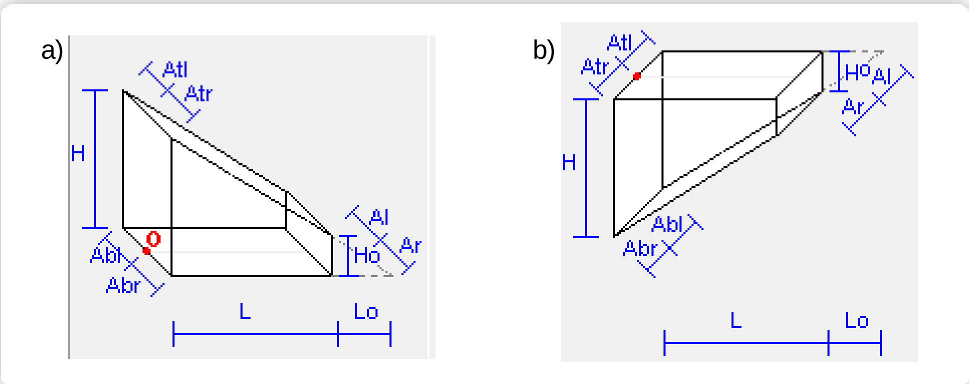


Fig. 12: Geometry parameters for dimensioning.
a) Cross arm with bottom horizontal girder; b) cross arm with top horizontal girder.

Comments

[DN] – (01.2015):

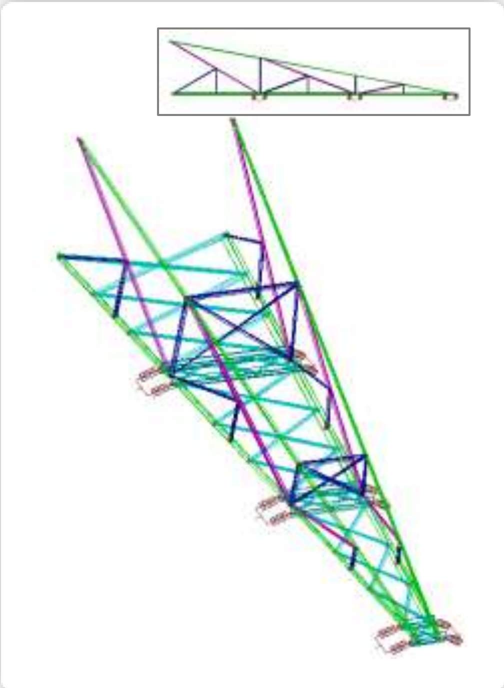


Fig. 13: Typical cross arm.
- 3 conductor attachment points
- 2 vertically bracings
- tip height of 0

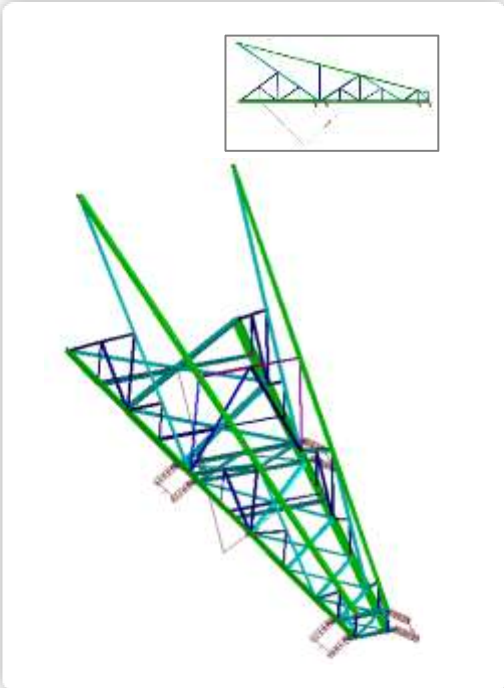


Fig. 14: Typical cross arm.
- 2 conductor attachment points and
- 1 V insulator with additional attachments
- 1 vertically bracing

Exceptions:

(Often) The tip height H0=0 (see sketches Fig. 12).

(Often) In case of more than one conductor member, the tip length L0 = b/2; b := distance between the outer conductor members (see Fig. 15).

(Sometimes) The tip width Ar=Al=0 (see sketches Fig. 12).

(Sometimes) The horizontal girder is the top girder (see Fig. 17).

(Rarely) The top girders have two different declinations (see Fig. 18).

(Rarely) The cross arm has no diagonal or brace or redundant elements in the panels of the vertical view, only corner members.

(Occasionally) The connection definitions of a rotated cross arm is not directly at points of the tower body, e.g. by an additional horizontal bracing, much larger than the tower body width, (see Fig. 22).

(Occasionally) The cross arm consists only of a horizontal girder.

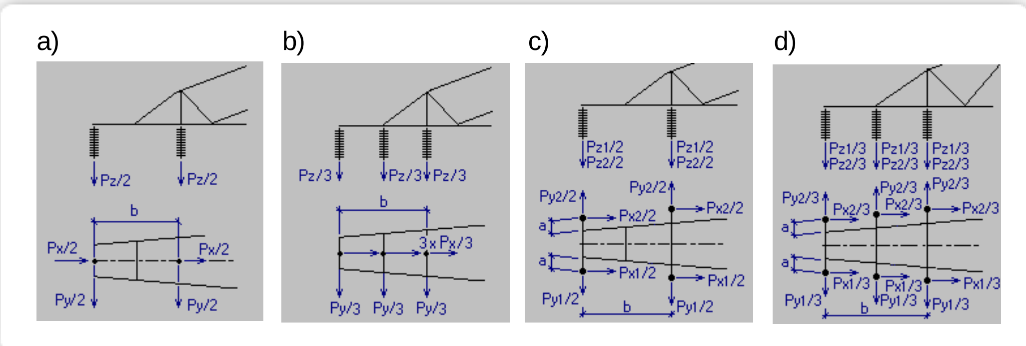


Fig. 15: Four suspension types with L0=b/2.

Comments

[DN] – (01.2015):

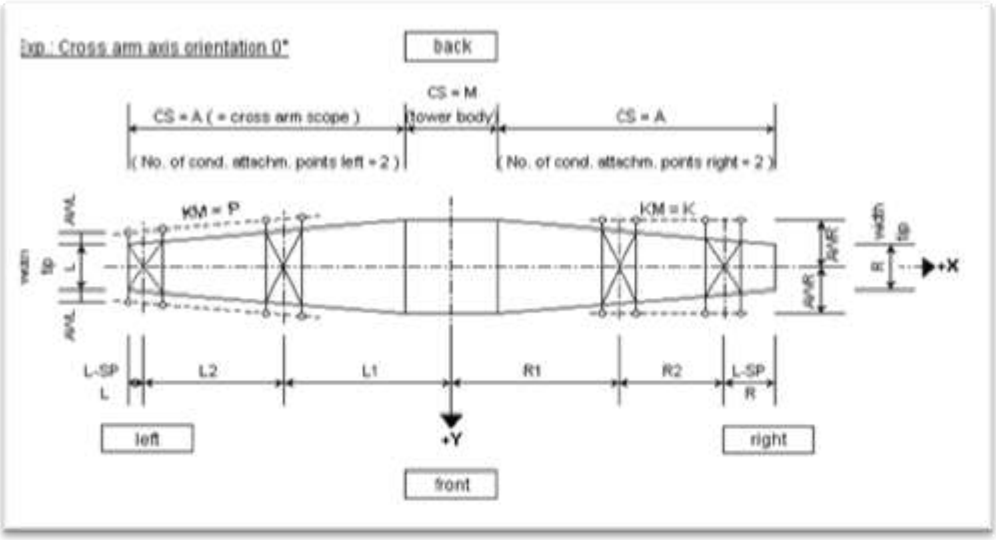


Fig. 16: Sketch: Definition of the conductor attachment points.

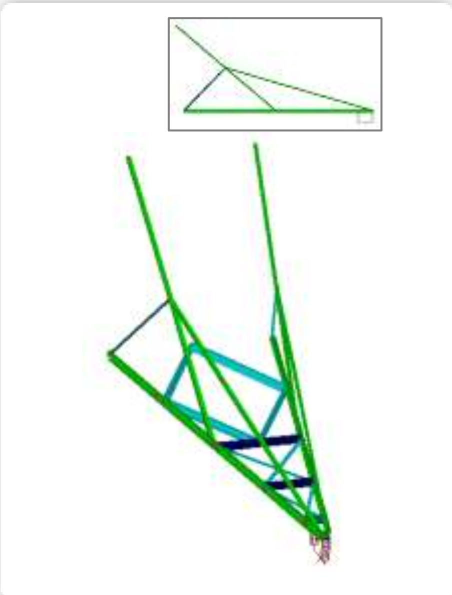


**Fig. 18: Cross arm with one conductor attachment point.
The upper girders have 2 different declinations.**

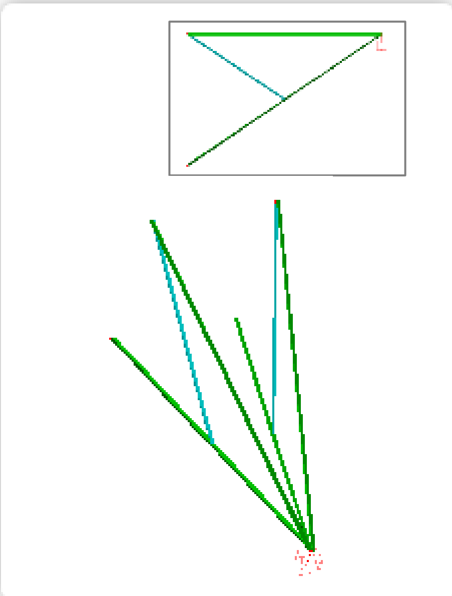


**Fig. 17: Simple cross arm construction with a horizontal upper girder.
All corner elements are connected in one point at the cross arm tip, which is the conductor attachment point too, without a conductor attachment member.**

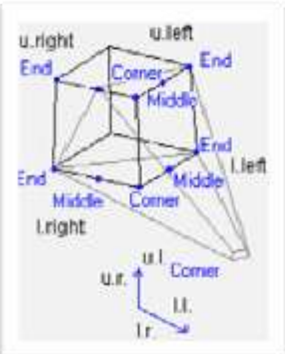


Fig. 19: Connection definitions for rotated cross arm.

Comments

[DN] – (01.2015):

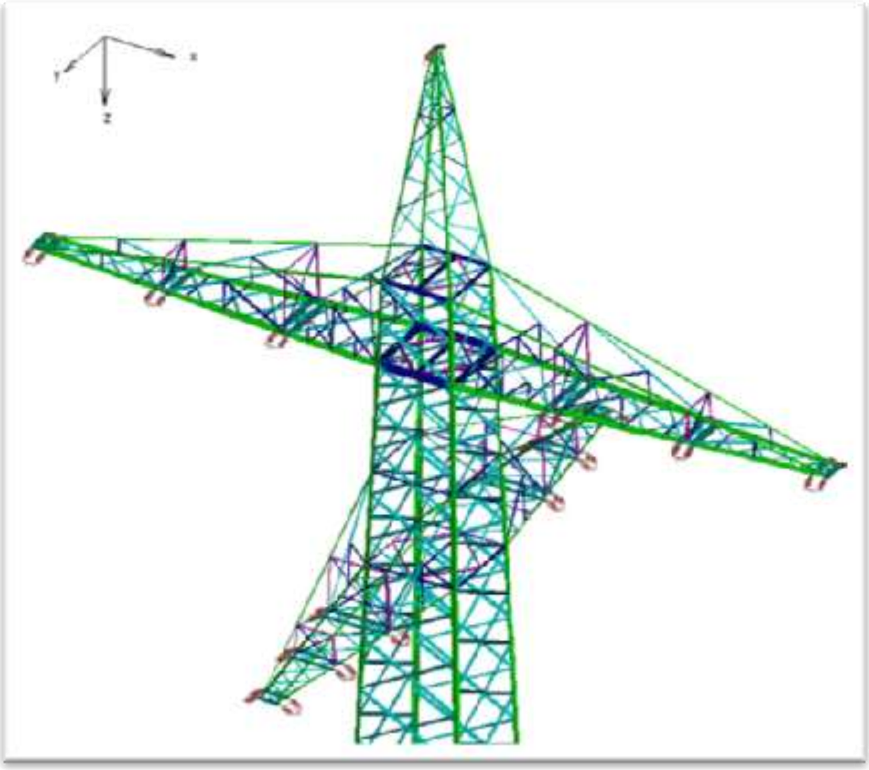


Fig. 20: Standard tower with X-cross arm and Y-cross arm at two levels

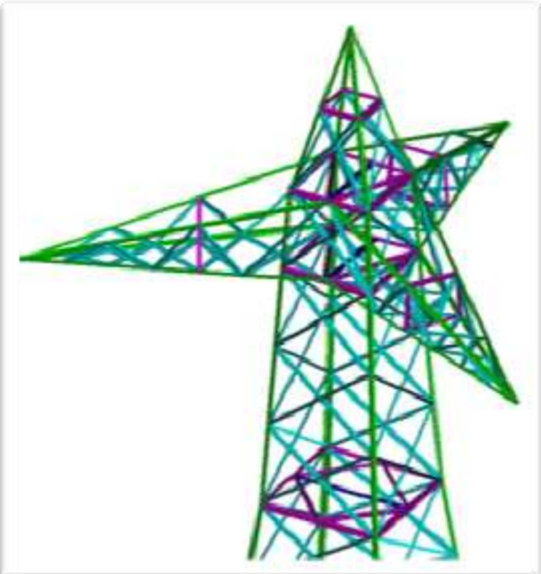


Fig. 21: Tower with three cross arms at one level, called „Mercedes cross arms”.

- **The two cross arms over corner are connected with the corner elements one side in the middle and on the other side at the end.**
- **All corner elements hit in one hinge at the cross arm tip, without a conductor attachment member.**

Comments

[DN] – (01.2015):

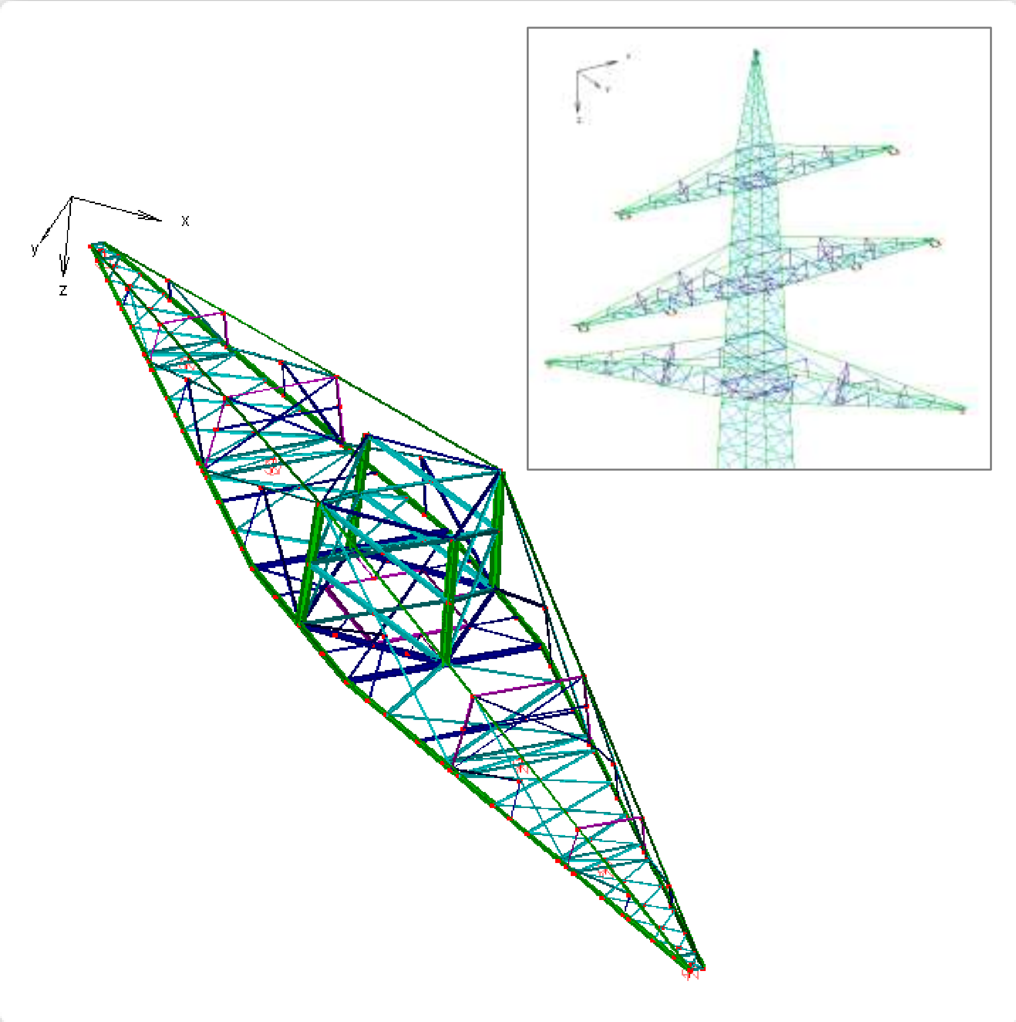


Fig. 22: Standard tower with two X – cross arms and one cross arm rotated by 45°.

- **The lower girders of the rotated cross arm are connected to the tower body at the corner points of an additional rotated and enlarged horizontal bracing.**
- **The upper girders are connected at the end points of the existing tower body bracing.**

Data Structure: Tree view

Basic tree view objects

- …… *“Project Name”*
  - …… Mainstructure
    - …… Tower Tops
    - …… Tower Body
    - …… Cross Arms
  - …… *“Name of Variant 1”*
    - …… Tower Body
  - …… *“Name of Variant 2”*
  - …… *“Name of Variant 3”*

⋮

Comments

[DN] – (01.2015):

Tree view substructure for “Tower Tops”

- …… Tower Tops
  - …… *“Name of Tower Top 1”* (eg. *“ES1”*)
    - …… Corner members
      - …… *“Name of Section 1”*
        - …… *“Name of Corner 1”*
        - …… *“Name of Corner 2”*
        - ⋮
      - …… *“Name of Section 2”*
    - …… Panels (*“Tower Top 1”*)
      - …… *“Name of Panel 1”* (eg. *“ES1_FB1”*)
        - …… Diagonal members
          - …… *“Name of Diagonal 1”*
          - …… *“Name of Diagonal 2”*
          - ⋮
        - …… Brace members
          - …… *“Name of Brace 1”*
          - …… *“Name of Brace 2”*
          - ⋮
        - …… Redundant members
          - …… *“Name of Redundant 1”*
          - …… *“Name of Redundant 2”*
          - ⋮
        - …… Conductor attachment members
          - …… *“Name of Conductor attachment 1”*
          - …… *“Name of Conductor attachment 2”*
      - …… *“Name of Panel 2”*
  - …… *“Name of Tower Top 2”*

Comments

[DN] – (01.2015):

Tree view substructure for “Tower Body”

- …… Tower Body
  - …… *“Name of BZ 1”* (eg. *“BZ – 65 mm/m”*)
    - …… Corner members
      - …… *“Name of Section 1”* (eg. *“S1”*)
        - …… *“Name of Corner 1”*
        - …… *“Name of Corner 2”*
      - …… *“Name of Section 2”*
    - …… Panels of Tower Body (*“BZ 1”*)
      - …… *“Name of Panel 1 of BZ 1”* (eg. *“S1_FB1”*)
        - …… Diagonal members
        - …… Brace members
        - …… Redundant members
        - …… Conductor attachment members
      - …… *“Name of Panel 2 of BZ 1”* (eg. *“S1_LR2”*)
      - …… *“Name of Panel 3 of BZ 1”* (eg. *“S1_B3”*)
  - …… *“Name of BZ 2”* (eg. *“BZ – 210 mm/m”*)
    - …… Corner members
      - …… *“Name of Section 1”* (eg. *“S4”*)
      - …… *“Name of Section 2”* (eg. *“S5”*)
    - …… Panels of Tower Body (*“BZ 2”*)
      - …… *“Name of Panel 123”* (eg. *“S4_LR123”*)
      - …… *“Name of Panel 123”* (eg. *“S4_FB123”*)
    - …… Panels of Leg Extension (*“BZ 2”*)
      - …… *“Name of Panel Front–Support A”* (eg. *“S7_F-A123”*)
        - …… Diagonal members
        - …… Brace members
        - …… Redundant members
      - …… *“Name of Panel Left–Support A”* (eg. *“S7_L-A123”*)
      - …… *“Name of Panel Left–Support D”* (eg. *“S7_L-D123”*)
      - …… *“Name of Panel Back–Support D”* (eg. *“S7_B-D123”*)

Comments

[DN] – (01.2015):

Tree view substructure for “Cross Arm”

- ...... Cross Arms
  - ...... *“Name of Cross Arm 1”*
    - ...... Upper Corner members
      - ...... *“Name of Section 1, Arm 1, right”* (eg. “*AR1_S1*”)
        - ...... *“Name of Upper Corner 1”*
        - ...... *“Name of Upper Corner 2”*
        - ⋮
      - ...... *“Name of Section 2, Arm 1, right”*
    - ...... Lower Corner members
      - ...... *“Name of Section 1, Arm 1, right”*
      - ...... *“Name of Section 2, Arm 1, right”*
    - ...... Panels of Cross Arm 1 (“*Front-Back*”)
      - ...... *“Name of Panel 1 of AR 1”* (eg. “*AR1_FB123*”)
        - ...... Diagonal members
        - ...... Brace members
        - ...... Redundant members
      - ...... *“Name of Panel 2 of BZ 1”* (eg. “*AR1_FB2*”)
      - ...... *“Name of Panel 3 of BZ 1”* (eg. “*AR1_FB3*”)
      - ⋮
    - ...... Panels of Cross Arm 1 (“*Lower-Upper*”)
      - ...... *“Name of Panel 1 of AR 1”* (eg. “*AR1_L1*”)
        - ...... Diagonal members
        - ...... Brace members
        - ...... Redundant members
        - ...... Conductor attachment members
      - ...... *“Name of Panel 2 of BZ 1”* (eg. “*AR1_L2*”)
      - ...... *“Name of Panel 3 of BZ 1”* (eg. “*AR1_U3*”)
      - ⋮
  - ...... *“Name of Cross Arm 2”*
  - ...... *“Name of Cross Arm 3”*
  - ⋮

Comments

[DN] – (01.2015):

To Pos. 1.5: Databases of FMS

Conductor Database (Leiter-Datenbank)

File type { “*.rop“ }. Include as is.

Conductor parameters:

Conductor diameter	22.4	[mm]
Conductor weigth	0.998	[kg/m]
Coeff of therm exp	0.196	*1E-04 [1/K]
Young's modulus	74	[kN/mm2]

Static calculation

Number of wires	24	
Wire diameter 1	3.74	[mm]
Cross-section 1	263.66	[mm2]
Number of wires	7	
Wire diameter 2	2.49	[mm]
Cross-section 2	34.09	[mm2]

Conductor material

Resistance	0.1094	[Ohm/km]
Ultimate force	82.94	[kN]

Other parameters

Additional data

Name extension	LS 565-AL1/72-ST1A	
No. of sub-conductors	4	
Tension	46.0	[N/mm²]
Corresponding temperature	10.0	[°C]
Ice load q i1	0.0	[N/m]
Wind load w1	0.0	[N/m]

Local parameters, not in the conductor db, but important for determining the loads for each conductor arrangement.

Remark: The shown values in the snapshots above are fictive only as an example.

Comments

[DN] – (01.2015):

Bolt Database (Schrauben-Datenbank)

File type { “*.blt“ }. Include, but reduced only for needed data values.

No.	Name	Bolt Type	Size	V.Type	Grade	fybk	fubk	d1	dL	d	A.s	A.sp	mue
	*	*	*	*	*	[N/mm²]	[N/mm²]	* [mm]	* [mm]	* [mm]	* [mm²]	* [mm²]	* [-]
31	M30 -10.9-SL	R	M30	SL	10.9	900	1000	30	1	30	707	561	0.5
32	M36 -10.9-SL	R	M36	SL	10.9	900	1000	36	1	36	1018	817	0.5
33	M12 -4.6-SLP	P	M12	SLP	4.6	240	400	13	0	13	133	84.3	0.5
34	M16 -4.6-SLP	P	M16	SLP	4.6	240	400	17	0	17	227	157	0.5

z.Sig.L H	z.Sig.L HZ	z.Tau a H	z.Tau a HZ	AlphaA
[N/mm²]	[N/mm²]	[N/mm²]	[N/mm²]	*
280	320	240	270	0.55
280	320	240	270	0.55
320	360	140	160	0.6
320	360	140	160	0.6

Fig. 23: Bolt parameters needed for the variant of maximal parameter count

The parameter ‘Bolt Type’ (3rd column) can be delete too, because this information is given also by the parameter ‘V.Type’ (5th column).

The parameters ‘z.Sig.L H’, ‘z.Sig.L HZ’, ‘z.Tau a H’ and ‘z.Tau a H’ are only relevant for old standard. Maybe that they are necessary sometime.

One parameter shall be added. ‘k2’ : parameter for the bolt tension design equation.

No.	Name	Size	V.Type	Grade	fybk	fubk	d1	dL	d	A.s	A.sp
	*	*	*	*	[N/mm²]	[N/mm²]	* [mm]	* [mm]	* [mm]	* [mm²]	* [mm²]
31	M30 -10.9-SL	M30	SL	10.9	900	1000	30	1	30	707	561
32	M36 -10.9-SL	M36	SL	10.9	900	1000	36	1	36	1018	817
33	M12 -4.6-SLP	M12	SLP	4.6	240	400	13	0	13	133	84.3
34	M16 -4.6-SLP	M16	SLP	4.6	240	400	17	0	17	227	157

Fig. 24: Bolt parameters needed for the variant of minimal set of parameters

In case of this variant, the parameters ‘mue’, ‘AlphaA’ and ‘k2’ as well as the parameters for old standard have to define individual for each design calculation of a tower project.

Panel Database (Feld-Datenbank)

File type { “*.fld“ }. Include, but only geometry data, without corner elements. The range of panel types can be reduced to one panel type, only the panel type ‘*Simple*’ with 4 or 3 corners is needed (see ‘*Panel Editor Modul*’ P27).

All design parameter definitions are set in the tower project directly to element groups and, if necessary, can be individually modified for each element.

Comments

[DN] – (01.2015):

Required are an import function of a FMS panel database and a writing routine of the geometry data to the new format of the GAME panel database.

A standard cross diagonal panel is given as an illustrative example:

Parameters of ‘Nodes’

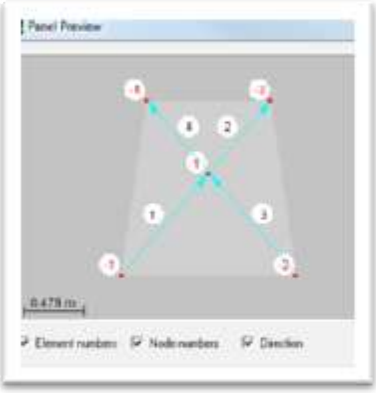


Fig. 26: Panel preview

Fig. 25: Old table for the node parameters

Beside the 4 corner nodes (-1, -2, -3, -4) only one new node, the crossing node has to be defined for the panel ‘B30’ of crossing diagonals.

To define additional nodes, the 5 types of nodes are sufficient:

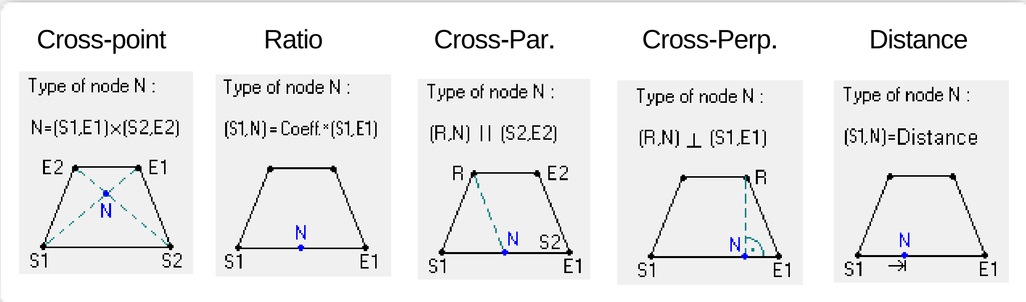
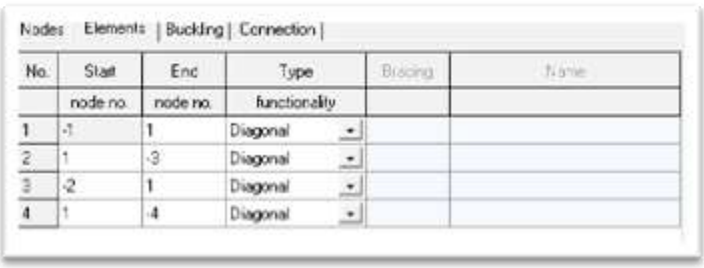


Fig. 27: Auxiliary node types

Parameters of ‘Elements’



No.	Start node no.	End node no.	Type functionality	Bracing	Name
1	-1	1	Diagonal		
2	1	-3	Diagonal		
3	-2	1	Diagonal		
4	1	-4	Diagonal		

Fig. 28: Old table for the element parameters

In the element parameter table, the parameters ‘Bracing’ (a check-button, active only for panel type ‘*Braced*’) and ‘Name’ (name of the bracing panel) become no longer necessary if the panel type ‘*Braced*’ is nonexistent.

The parameter ‘Type functionality’ comprised ‘*Diagonal*’, ‘*Redundant*’, ‘*Brace*’ (‘*Diagonale*’, ‘*Sekundärfachwerk*’, ‘*Verband*’). In case of defining the corner members outside the panel definition, the element functionality type ‘*Corner*’ (‘*Ecke*’) can be omitted here.

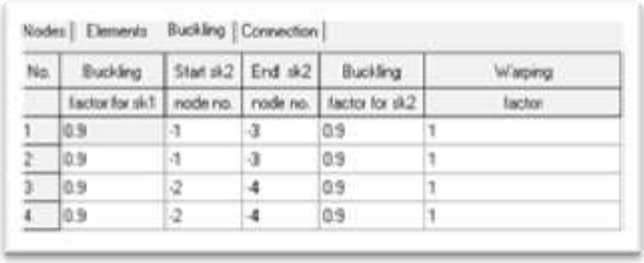
Comments

[DN] – (01.2015):

Comments

[DN] – (01.2015):

Parameters of ‘Buckling’



Nodes | Elements | Buckling | Connection

No.	Buckling factor for sk1	Start sk2 node no.	End sk2 node no.	Buckling factor for sk2	Warping factor
1	0.9	-1	-3	0.9	1
2	0.9	-1	-3	0.9	1
3	0.9	-2	-4	0.9	1
4	0.9	-2	-4	0.9	1

Fig. 29: Old table for the buckling parameters

The buckling parameters rank not only among design parameters, they are geometry parameters too. We distinguish between buckling about minor section axis (usually element length), index 1, and buckling perpendicular to panel plane (overall member length), index 2.

The factor parameters for sk1 and sk2 describe percentage the influence of reduced element length due to the distance from the element end node position of the static model to the center point of a bolted connection (see Fig. 30).

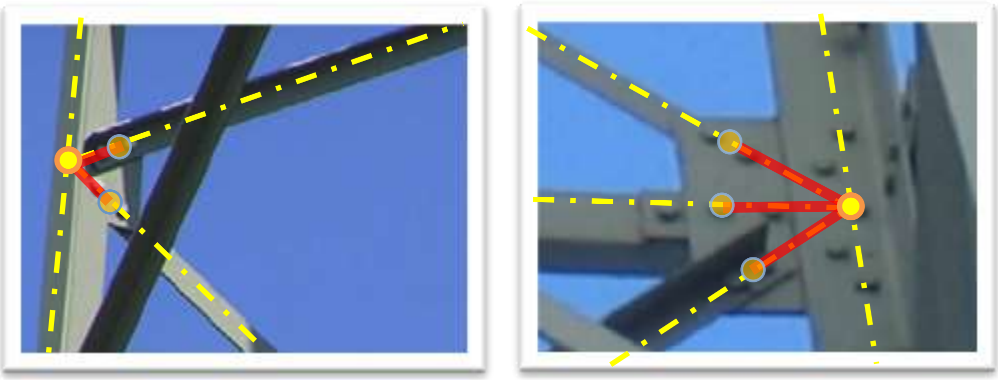
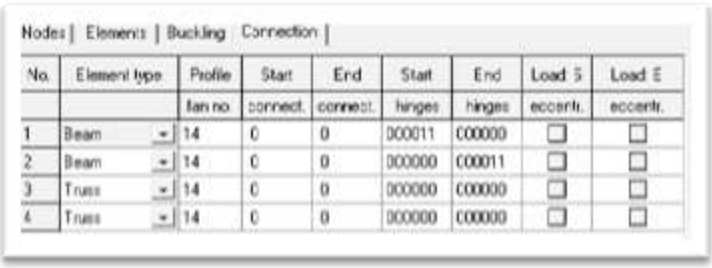


Fig. 30: Illustration of the buckling length reduction
static model of element axes and nodes
buckling length reduction due to the position of the center point for the bolted connections

The parameter ‘Warping factor’ describes percentage the warping length reduction likewise the buckling factors. This factor was not used in the FMS program for the default design standard, but shall be not eliminate for GAME.

Parameters of ‘Connection’



Nodes | Elements | Buckling | Connection

No.	Element type	Profile fan no	Start connect.	End connect.	Start hinges	End hinges	Load S eccentr.	Load E eccentr.
1	Beam	14	0	0	000011	000000	☐	☐
2	Beam	14	0	0	000000	000011	☐	☐
3	Truss	14	0	0	000000	000000	☐	☐
4	Truss	14	0	0	000000	000000	☐	☐

Fig. 31: Old table for the connection parameters

It makes no sense to define the parameters of the register ‘*Connection*’ in the panel editor and store them into the panel database. Furthermore the two parameters ‘Start connect.’ and ‘End connect.’, originally planned for the number of the conformed bolt fan, are not used in the program FMS. Nevertheless, all 8 parameters belong to the element properties.

Profile Database (Profil-Datenbank)

File type { “*.mdb“ }. Include as is, with 2 exceptions.

(1) Reducing the types of profile databases:

- *sstn_std.mdb*: standard profile database, read only, one DB for all users;
- *sstn_grp.mdb*: group profile database, read/write, for materials and profiles which are not given in the standard profile DB but often used, preferred one DB for all users;
- ~~*sstn_usr.mdb*~~: user profile database, read/write, not any more necessary;
- *sstn_prj.mdb*: project profile database, read/write, local DB apply to project, includes special profile definitions;
- *sstn_Empty.mdb*: template profile database, read/write, needed for generating e.g. a project profile database.

(2) The access to profile parameters from running program GAME

The profile parameters are reading once from a profile DB and store them to the input file of the project and not each time if they are used in the program. This minimizes the errors in case of accessing to a profile or a material of profile DBs, with different parameters of equal named profiles or materials.

Profile Fan Database (Profilfächer Datenbank)

File type { “*.pfn“ }. Is useful and shall be converted as is.

Bolt Fan Database (Schraubenfächer Datenbank)

File type { “*.bfn“ }. Program users not often use this file type.

The bolt sets are generated easy and quick, because of less number of bolts in a list, often only one bolt, and the bolt sets of a BFN-file are in general only for one project usable.

Bolt sets are needed, but not a separate database.

Wind Drag Database (Windstaudruck Datenbank)

File type { “*.wnf“ }. The wind drag sets are needed, but the sets are generated in Excel-sheets and copied from Excel directly to the FMS dialog, therefore the user use very rarely the WNF-file DB.

In the FMS program the wind drag curves are defined as pairs of variants (Level – Drag) not as an equation. These wind drag sets can be easy build in Excel-

Comments

[DN] – (01.2015):

sheets and also store as Excel-files. Therefore the drag/drop functionality of values from Excel-sheet to the 'Wind Drag Function' dialog shall be not omitted and is important.

Remark: In FMS there is only one wind drag set possible for all load cases with wind loads. GAME shall offer the use of several wind drag sets for one tower project.

Wizard Data File (Mast-Wizard Datei)

File type { "*.mwd" }. There is no need of this data set and therefore it is no need to include it into GAME. Reason: The tower type data set is part of the input file itself.

Exception:

In the FMS program 5 tower types are available:

- "Standard Tower" (Standardmast)
- "Standard tower with two tops" (Standardmast mit geteilter Erdseilspitze)
- "Portal Tower" (Portalmast)
- "Y-Tower: Bull" (Y-Mast: "Bull")
- "Y-Tower: Cat" (Y-Mast: "Cat")

In GAME only the first two tower type's need selectable, the other three tower types are not frequented very often. Therefore it makes no sense to begin the program development with all 5 tower types and investigate much time for implementation of source code for it. If somebody need to define a tower of these three types, than he can use the FMS program.

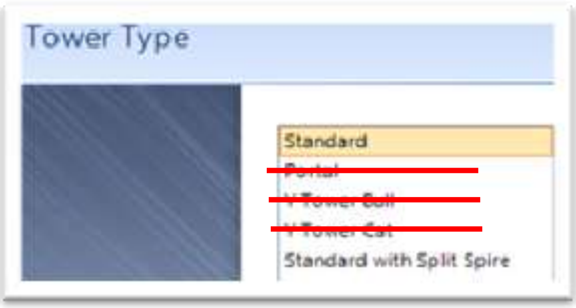


Fig. 32: List of tower types

Comments

[DN] – (01.2015):

RGL File (Regel Datenbank Datei)

File type { “*.rgl“ }. Not included as is.

This file enclose parameters, factors and generalized rules to determine loads, load combinations, load cases, stages of conductor erection, etc. and to execute the design calculation. This set of data offer to the user a solid base and a compendium for the complex analysis of loading and management of design.

The import of this templates of rules enabled a very efficiently and economic procedure due to the execution of tower calculation.

A draft for this rule database can defined first, if the data structure for the loads, load cases and load combinations exists.

Basic Block Database (Grundkörper Datenbank)

File type { “*.blk“ }. Not clear, but possible not included.

The FMS program can store besides the parameters for a basic block itself also the complete substructure including all properties for the panels and the members, if available. For the program GAME basic blocks are only a graphical tool to build easy a 3D tower model and therefore differ to the basic block in FMS. An import of BLK-file data is not needed.

If the program GAME presents clear substructures likewise a cross arm half or a tower body top or a tower body main structure etc., than it make sense to have the possibility to store such a substructure in a separate file and to load (import) this set of data into another tower project.

Comments

[DN] – (01.2015):

To Pos. 1.6: Redefining the editor modules

Bolt Editor Module (old name: “*TOWER schrauben*”; new name: ?)

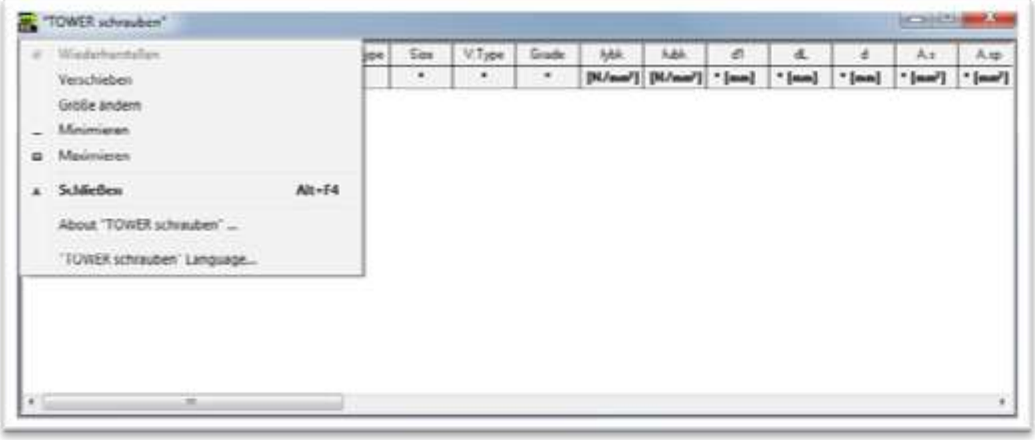


Fig. 33: Main window of the bolt-editor module for the FMS program

The main window of the old bolt editor is a table, simple and effective. The number of columns can be reduced, because several values are not needed and explode only the table width.

Suggestion: Equal set of data as in old editor. But only those columns for the minimum set of parameters are shown for the standard edit mode (see also “*Bolt Database (Schrauben-Datenbank)*”). If the user is interested to store more information’s for a new bolt input, than he can activate all columns. This reduces source code development to handle compatibility problems.

Parameters are:

```
CString m_sName;          // Name of the bolt

int     m_nNorm;          // index to m_NormArray;
int     m_nGroup;         // index to m_GroupArray;
CBUnitType  m_eUnit;          check of need
CBBoltType  m_eBolt;
int     m_nBoltSize;      // index to m_BoltSizeArray;
CBConnectionType  m_eConnection;
int     m_nClass;         // index to m_ClassArray;

double  m_dfybk;          // Streckgrenze[N/mm^2]                    > 0.0
double  m_dfubk;          // Zugfestigkeit [N/mm^2]                  > 0.0
double  m_dd1;            // Gewindedurchmesser d1 [mm]              > 0.0
double  m_ddL;            // Lochspiel [mm]                          >= 0.0
double  m_dd;             // Schaftdurchmesser d [mm]                > 0.0

int     m_nNutNorm;       // index to m_NutNormArray
double  m_dk;             // Kopfhöhe k [mm]                         > 0.0
double  m_dm;             // Mutterhöhe m [mm]                       > 0.0
double  m_ds;             // Schlüsselweite s [mm]                   > 0.0
double  m_deEck;          // Eckenmaß min e [mm]                     > 0.0

int     m_nWasherNorm;    // index to m_WasherNormArray
double  m_dsd;            // Scheibendurchmesser d.Sch [mm]          > 0.0
double  m_dst;            // Scheibendicke t [mm]                    > 0.0
double  m_dKlmin;         // min Klemmlange [mm]                     > 0.0
double  m_dKlmax;         // max Klemmlange [mm]                     > 0.0
double  m_dASch;          // Schaftquerschnittsfläche A.Sch [mm^2]   > 0.0
double  m_dASp;           // Spannungsquerschnittsfläche A.Sp[mm^2]  > 0.0
```

Comments

[DN] – (01.2015):

```
double  m_dAk;           // Kernquerschnittsfläche A.k [mm^2]     > 0.0
double  m_dmue;          // Reibungszahl     [ ]                  > 0.0
double  m_dFv;           // Vorspannkraft Fv [N]                  > 0.0
double  m_dzulSigmaL_H;     // zul Sigma.L fuer H   [N/mm^2]      > 0.0
double  m_dzulSigmaL_HZ;    // zul Sigma.L fuer HZ  [N/mm^2]      > 0.0
double  m_dzulTau_a_H;      // zul Tau.a fuer H     [N/mm^2]      > 0.0
double  m_dzulTau_a_HZ;     // zul Tau.a fuer HZ    [N/mm^2]      > 0.0
```

```
//spring washer (Federringe):
int     m_nSpringNorm;      // FederringnormBei."DIN 127" - index of array;
int     m_nSpringSize;      // Federring Nenngröße("A 18") - index of array;
double  m_dFsd;             // Federringdurchmesser d.Sch [mm]    > 0.0
double  m_dFst;             // Federringdicke t  [mm]             > 0.0

//Bolts: Lochspiel
double  m_dGrenzB;          // Größte Schenkelbreite Grenz b ?    > 0.0
```

not used

```
//Zul Abstände e, e1, e2 ?
double  m_dzul_e;           // Zul Abstand e                      > 0.0
double  m_dzul_e1;          // Zul Abstand e1                     > 0.0
double  m_dzul_e2;          // Zul Abstand e2                     > 0.0
double  m_dzul_e3;          // Zul Abstand e                      > 0.0
```

delete

```
double  m_dAlphaA;          // Beiwert a für Abscherfestigkeit    > 0.0

// design stresses according american standard ASCE
double  m_dSigma_ASCE;      // sigma for ASCE [N/mm^2]            >= 0.0
double  m_dTau_ASCE;        // tau for ASCE [N/mm^2]              >= 0.0
```

Remark:

For the bolt tension design, parameter '**k_2**' is missing in the old data set.

Panel Editor Module (old name: "*TOWER felder*"; new name: ?)

The old panel-editor starts with a main window for selecting a panel database file **{ "*.fld" }** and shows the panels for the selected database in a list.

For GAME only the panel type "***Simple***" with 4 or 3 corners is needed. The other more complex types as "***Double***", "***Brace***" and "***Hip***" could be ignored. At present there couldn't present a solution how the parameters of these panel types could be converted to the panel type "***Simple***".

The parameters for '***Nodes***', '***Elements***' and '***Buckling***' are needed up to the two parameters 'Bracing' and 'Name' (see Fig. 28, both as a part of register 'Elements'), see "*Panel Database (Feld-Datenbank)*" chapter too.

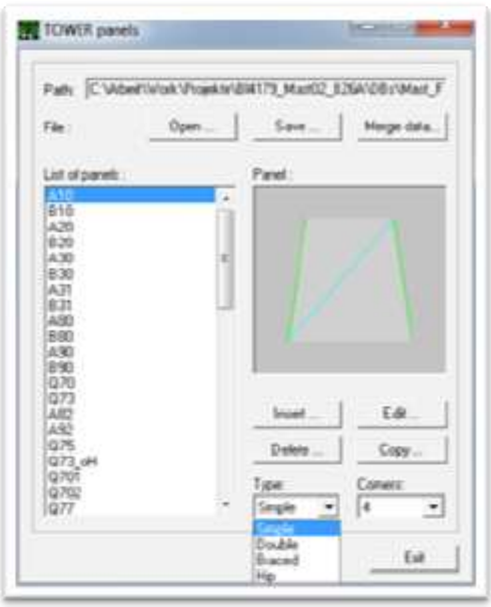


Fig. 34: Main window of ‚TOWER panels' editor modul

Comments

[DN] – (01.2015):

The parameters of the register ‘Connection’ (see Fig. 31) could be shifted from the panel data set. The definition of ‘Element type’, ***Beam*** or ***Truss***, ‘Profile fan no.’, the degrees of freedom for ‘Start hinges’ and ‘End hinges’, and if there is an eccentricity of the normal force at the start and/or end node, ‘Load: S. eccentr.’ ‘Load: E. eccentr.’, these are element properties and not panel properties.

Conductor Editor Module (old name: “*TOWER leiter*”; new name: ?)

The old conductor editor modul opens with a main window for selecting a conductor database file **{ “*.rop“ }** and shows the panels for the selected database in a list., likewise as the old panel editor,

All parameters of the conductor data set are included into the new conductor data set see “*Conductor Database (Leiter-Datenbank)*”.

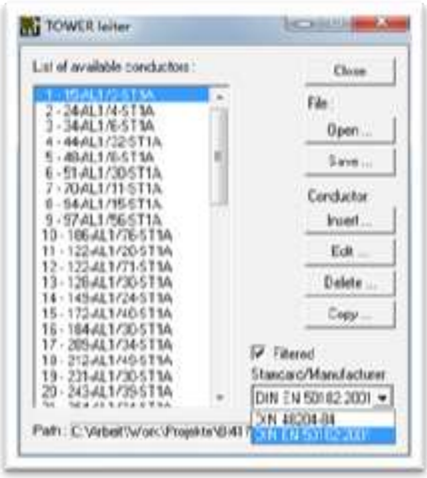


Fig. 35: Main window of ‚TOWER leiter‘ editor modul

Parameters are:

```
//
  int      m_nNumber;             // list number of the conductor
  CString  m_sName;               // name of the conductor, defined by user
  int      m_nNorm;               // standard/manufacturer, filter setting
// parameters for the static calculation
  double   m_dDiameter;           // conductor diameter [mm] (>=0)
  double   m_dMass;               // conductor weigth [kg/m] (>=0)
  double   m_dThermExp;           // coefficient of thermal expression *1e-4[1/K]
  double   m_dEModul;             // young's modulus [kN/mm²]
// parameters for the conductor material
  int      m_nCountWires1;        // number of wires [-]
  double   m_dDiameterWires1;     // wire diameter 1 [mm]
  double   m_dAreaWires1;         // cross section 1 [mm²]
//
  int      m_nCountWires2;        // number of wires [-]
  double   m_dDiameterWires2;     // wire diameter 2 [mm]
  double   m_dAreaWires2;         // cross section 2 [mm²]
// other parameters
  double   m_dDCResist;           // resistance [Ohm/km]
  double   m_dTensionCapacity;    // ultimate force [kN]

  CString  m_sParam16;            // not used
  CString  m_sParam17;            // not used
```

not used

Comments

[DN] – (01.2015):