Structural Analysis Software

Reference Manual

FEM48

Version 5.3

COPYRIGHT AND DISCLAIMER OF WARRANTY

CONTENTS

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Scope

FEM48 is THE PREMIER STRUCTURAL ANALYSIS SOFTWARE for the Hewlett Packard G series calculators. It's ease of use, flexibility, power and capabilities combined in one package are not seen in any other calculator program available, either commercially or as freeware.
The FEM48 library uses the Finite Elements Method and is designed to be used for the structural analysis of 2D frames, trusses and (continious) beams.

Main features

- modular approach
- extremely easy to use due to excellent interface
- supports frames, trusses and (continious) beams
- load cases and combinations are possible
- beam analysis (numerical and/or graphical)
- stress analysis (shear, combined axial and bending, Huber-Hencky)
- data can be entered in three (!) ways:
    - using a structure wizard for "standard structures"
    - using input prompts for structure data
    - assemble data arrays and store them from the stack
- text file (string) output of input and calculation results
- (un)deformed structure plots with optional displaying of supports, node numbers, members numbers
- seven (!) load types, including displacement loads, temperature loads and moving loads
- configurable matrix viewer/editor (choose your own matrix editor i.e. MATRIX or EDITB with Metakernel)
- configurable stringviewer (i.e. VV, the author's VIEW library or EDITB with Metakernel)
- superfast Cholesky matrix solve routine (assembly language)
- links to an external section database (SED48 v1.2 or higher)
- file management
- completely programmable (write your own batch files)

1.2 Method

FEM48 uses the displacement method for primary analysis. All other calculated values are derived from the nodal displacements.

1.3 Modules

FEM48 follows a modular approach. You can choose to load only the modules you need. The available modules integrate seamlessly into the FEM48 interface. Below a short description of each module is given. See the sections describing all behaviour of the modules (4, 5, 6, 7, 8, 9) for more details.
Each module is available in two versions[1], a "standard" version and a "compressed" version. The compressed versions are unique in that they provide exactly the same features as the standard versions at a smaller size! This because subroutines of these modules have been compressed. They decompress during runtime (which slows operations down a little bit). The standard and compressed modules can be mixed and matched. This allows you to select standard versions for much used modules and compressed versions for not so much used modules.

1.3.1 FEM48 module

This is the main module, which should always be installed. It provides the means of entering and editing data, calculation of the structure, plot routines and file management. The calculation only provides Finite Element Analysis, thus only data at the nodes will be calculated. For beam analysis you will need the QUERY module (see below).

[1] Compressed modules are not available for the HP 49G / 49G+ / 50G calculators.

1.3.2 QUERY module

This module is optional and provides beam analysis facilities. The member(s) to be analysed can be selected after which numerical and graphical output of axial force, shear force, bending moment, rotation, axial displacement, transverse displacement, shear stress, combined axial and bending stress and Huber-Hencky stress can be obtained.

1.3.3 WIZRD module

This module is optional and provides wizards for entering geometry and properties of "standard" structures like beams, bay frames and lattices. A link to SED48, a Structural Engineering Database by the author of FEM48 is provided.

1.3.4 PRINT module

This module is optional and provides commands for easy viewing and printing[2] of the input and output of FEM48.

1.3.5 MOVLD module

This module is optional and provides a command for generating moving loads along selected members of the current structure.

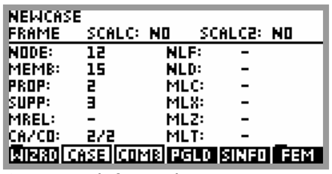
1.3.6 LCASE module

This module is optional and provides commands for managing load cases and load combinations.

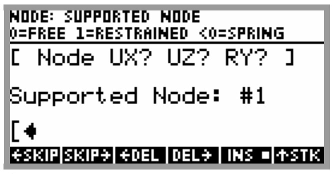
1.4 Screenshots

Below some screenshots of FEM48 in action are given to get a quick impression of the program.

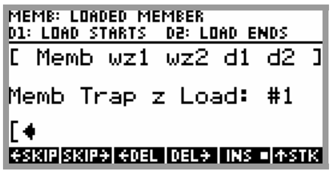
File management



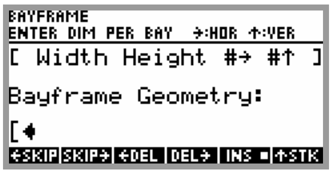
Structure information



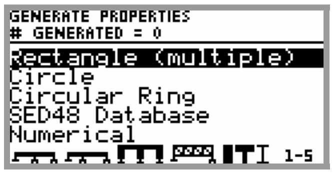
Supports entry prompt



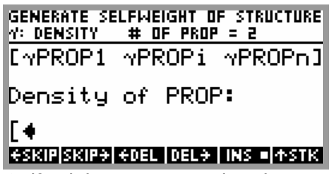
Trapezoid loads entry prompt



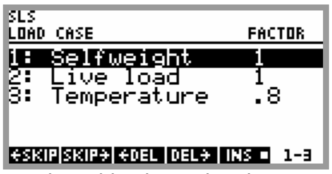
Bayframe geometry wizard



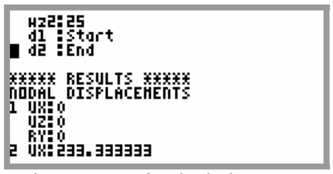
Section properties wizard



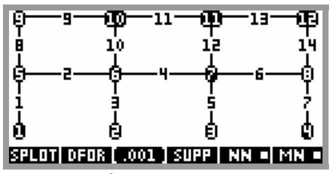
Selfweight generator wizard



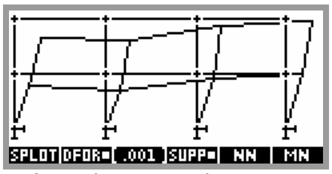
Load combinations wizard



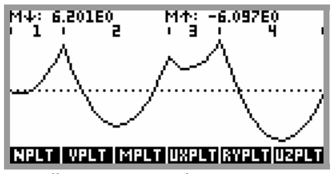
String output of calculation



Structure plot



Deformed structure plot



Bending moment plot

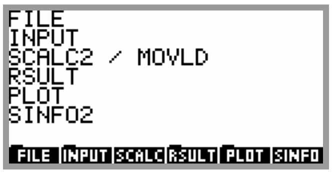
[2] If you own an HP calculator printer. Note that it is also easy when using FEM48 on an emulator, this enables you to copy/paste the strings to a text editor on your PC.

1.5 Menu structure

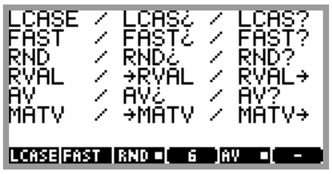
Below the complete menu structure of FEM48 and its modules is shown. Please note that most menukeys have actions in 3 planes! All keyactions are shown below: **unshifted / left-shifted / right-shifted**

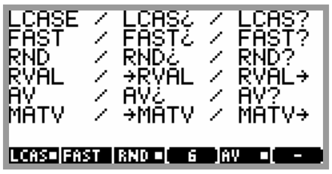
FEM menu — **LCASE inactive** / **LCASE active**

row 1



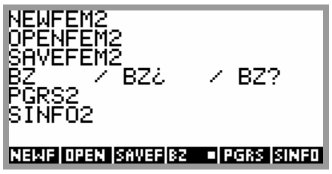
row 2





FILE menu — **LCASE inactive** / **LCASE active**

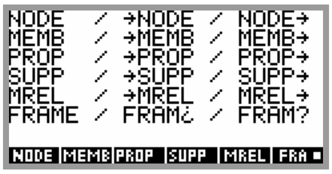
row 1

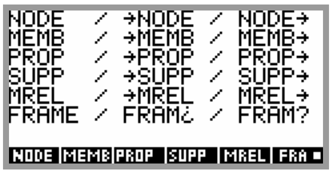


row 2

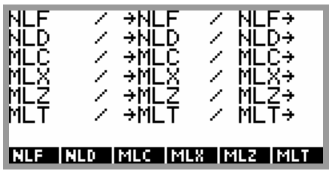
INPUT menu — **LCASE inactive** / **LCASE active**

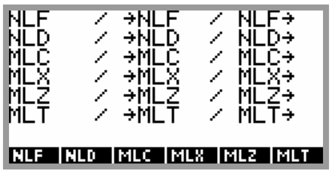
row 1





row 2



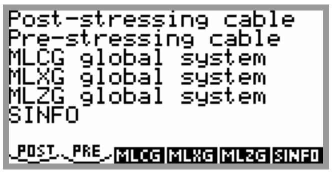


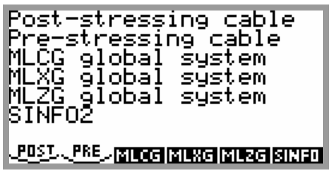
row 3

WIZRD menu | **LCASE inactive** | **LCASE active**

row 1

row 2



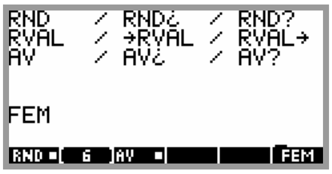


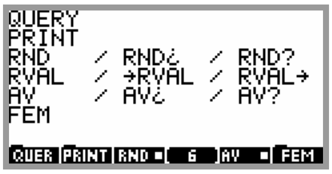
row 3

RSULT menu | **LCASE inactive** | **LCASE active**

row 1

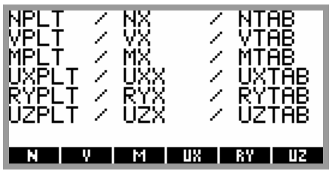
row 2

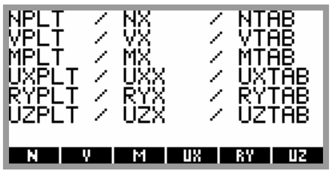




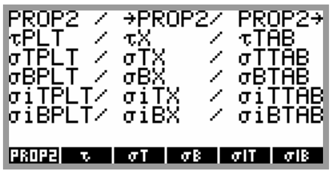
QUERY menu | **LCASE inactive** | **LCASE active**

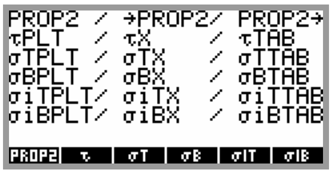
row 1





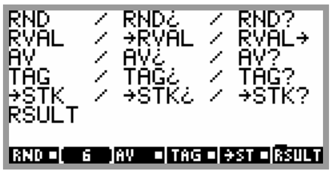
row 2

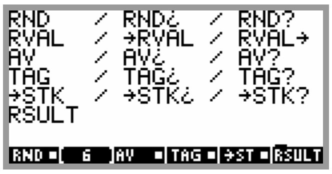


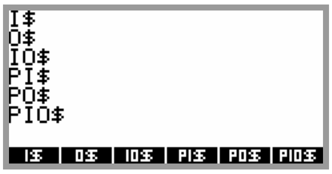
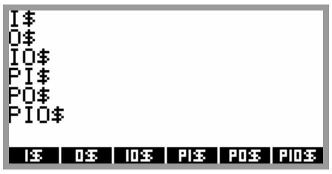
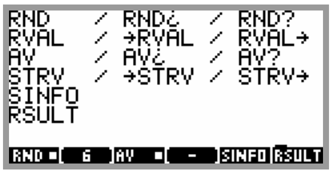
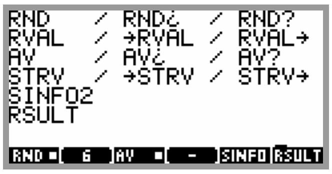


row 3

row 4





PRINT menu	**LCASE inactive**	**LCASE active**
row 1		
row 2		

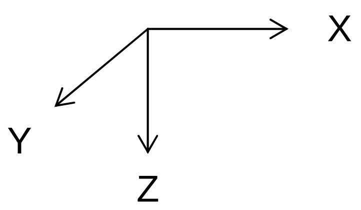
PLOT menu	**LCASE inactive**	**LCASE active**
row 1		
row 2		

2. COORDINATES AND UNITS

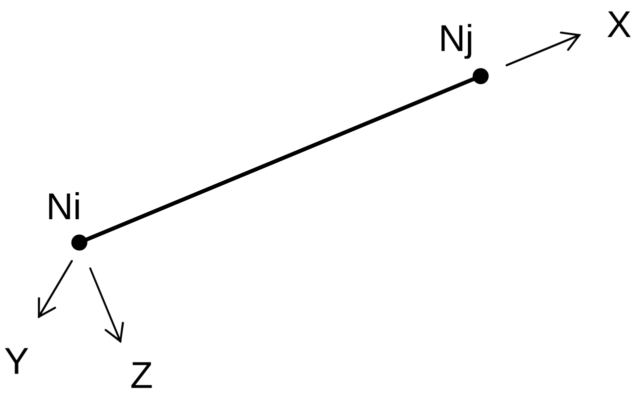
2.1 Coordinate systems

FEM48 uses a right hand rule X-Y-Z coordinate system where the positive X-direction is to the right and the positive Z-direction is downwards. Note that there are actually two coordinate systems. The global coordinate system and the local coordinate system.
The global coordinate system is used to enter the nodal coordinates. It is also used to enter nodal loads and to calculate nodal displacements.
The local coordinate system of a member is defined by the location of the start (Ni) and end (Nj) node of the member. It is used to enter member loads and to calculate member end forces.
<u>Please note that positive loads, reactions and displacements also follow these coordinate systems.</u>

Global coordinate system:



Local member coordinate system:



2.2 Units

Since the HP48 does not support units within arrays you have to be careful about the implied units you use. They must be consistent for meaningful results. So, if you use meter as the unit for length and kilo Newton as the unit for force you have to use square meter as the unit for area and kilo Newton meter as the unit for moment and so on.

3. QUICK START GUIDE

3.1 Data entry

Generally speaking, there are three ways to enter data with the FEM48 library:

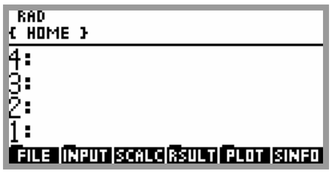
1. Use the inputline data prompts.
2. Assemble your own data matrices and store them with the →… commands.
3. Use the geometry, property and load wizards of the WIZRD module.

Below only method 1 is handled, have a look at the Command reference of the modules for more information about the other two data entry modes. Note that LCAS mode is <u>not</u> active during the following example.

3.2 Example of structure entry

Note: A general understanding of the HP48 calculator is assumed. Only the FEM48 module is discussed here. Please refer to the Command reference of the modules (4.3, 5.5, 6.3, 7.3, 8.3, 9.4) for more details.

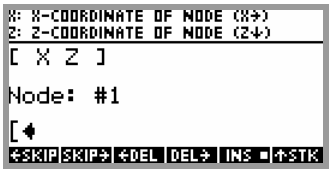
- Execute the FEM command (4.3.1)
  Now you have access to all commands with an easy to use interface.



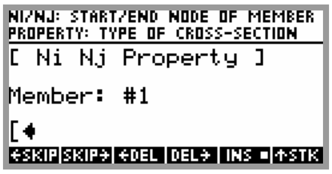
- Go to the INPUT menu
  Here all structure and load data can be entered and/or edited. Pressing a data menukey unshifted starts an inputline prompt for data, pressing it left-shifted stores data from the stack and pressing it right-shifted recalls the data to the stack (depending on the state of the AV¿ (4.3.58)*).*

About the inputline prompts described next: The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>. See the Command reference of each module for related commands with which you can edit data or enter data directly from the stack.

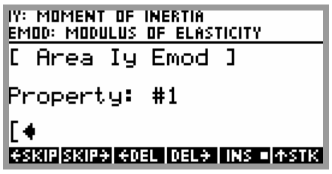
- Enter nodes with the NODE command (4.3.13).
  Note the coordinate system: X → and Z ↓.



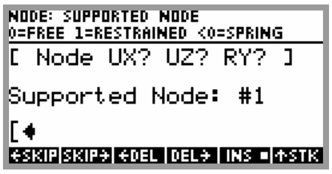
- Enter members with the MEMB command (4.3.19).
  Each member has a reference to a PROP, see below.



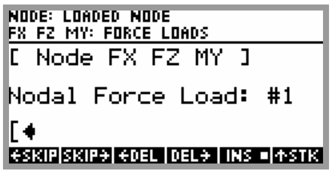
- Enter member properties with the PROP command (4.3.16).



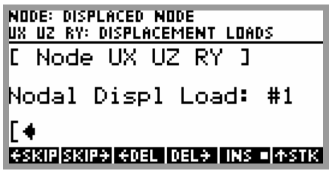
- Enter supports with the SUPP command (4.3.22).
  Springs can be entered too, just enter the spring value as a negative number.



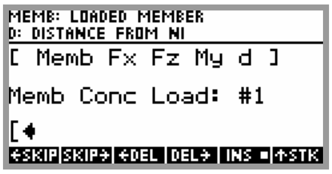
- Enter either one or more loads with the following commands:
  - NLF (4.3.28)
    Nodal force load (global system).



  - NLD (4.3.31)
    Nodal displacement load (global system).

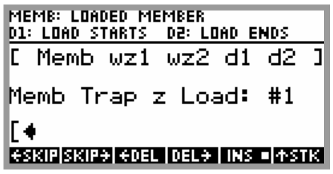


  - MLC (4.3.34)
    Concentrated member load (local system).



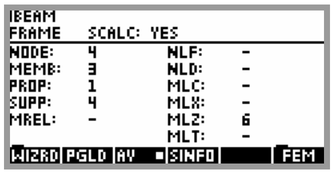
  - MLX (4.3.37)
    Trapezoid axial member load (local system). For uniform loads wx1 = wx2.

- MLZ (4.3.40)
  Trapezoid transverse member load (local system). For uniform loads wz1 = wz2.

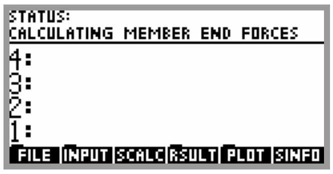


- MLT (4.3.43)
  Temperature member load. T_{axi} yields axial strain/stress, T_{diff} yields curvature/bending stress.

- Set the desired structure type with the FRAME toggle key on the 1st menurow, or use the FRAM¿ (4.3.52) command. *The default state of a new structure is FRAME.*
- Check what you have entered with the SINFO (4.3.7) command

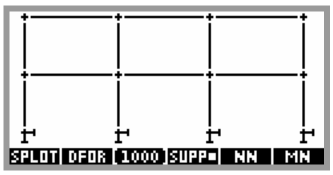


- Go back to the FEM menu and perform the calculation with the SCALC (4.3.47) command.



- Go to the RSULT menu.

- Look at the calculation results with one or more of the following commands:
  - NDIS (4.3.49)
    Nodal displacements (global system)
  - REAC (4.3.50)
    Support reactions (global system)
  - MFOR (4.3.51)
    Member end forces (local system)
- Go back to the FEM menu, go to the PLOT menu and use the SPLOT (4.3.48) command to plot the structure. *Also possible before calculation as a geometry check.*



- When the main calculation is finished, you can perform member analysis with the QUERY module, show results in string format (or print them) with the PRINT module etc, etc.

4. FEM48 MODULE

4.1 Description

The FEM48 module is the main module of the FEM48 library. This module has to be installed[3] in order to use other modules, since the other modules also use subroutines from the FEM48 module.
The FEM48 module provides the basic tools for structure analysis like file management, entering and editing of data, calculation of the structure and structure plot routines.

4.2 Menu structure

4.2.1 Library menu

After displaying the library menu (RS 2) and pressing the FEM48 library menukey you will see the FEM48 library menu. All commands of the FEM48 module are programmable and can be accessed from the library menu. However, this is not the best possible interface for easy use of FEM48. Therefore the FEM menu has been made, accessible through the FEM command.

4.2.2 FEM menu

The FEM menu, accessible through the FEM command provides a very easy to use interface for the FEM48 library. All commands of the FEM48 module[4] and the QUERY, WIZRD, PRINT, MOVLD and LCASE modules are accessible through this menu. See 1.5 for a complete menu description of all FEM48 modules.

4.3 Command reference

4.3.1 FEM

Displays the FEM menu.

Level 1	→	Level 1
	→	

Remarks:
When the LCAS¿ (4.3.54) option is enabled and the LCASE (9) module is present the FEM2 (9.4.2) command from the LCASE module is called. Suggested user key binding: 73.3 (RS 5)

4.3.2 CKFEM

Displays installed FEM48 modules and their version number.

Level 1	→	Level 1
	→	

Remarks:
All modules must be of the same version number as the main FEM48 module or they will be disabled (including programmable use). Only accessible from the FEM48 library menu.

4.3.3 ERASEFEM

Erases all hidden FEM48 files from the hidden directory[5] after confirmation.

[3] The commands of the other modules are protected and can not be used without the FEM48 module present or when this has a different version number.
[4] With the exception of the CKFEM, ERASEFEM and FBROW commands.
[5] A nullnamed subdirectory of HOME.

Level 1	→	Level 1
	→	

Remarks:

You will lose all structure data that has not been saved as a file. Only accessible from the FEM48 library menu.

4.3.4 NEWFEM

Starts a new "empty" structure.

Level 1	→	Level 1
	→	

Remarks:

If the current structure is not "empty" you will be prompted to save the current structure before it is replaced by an "empty" one.

4.3.5 OPENFEM

Opens a previously saved structure. You can select the file to open with a browser interface alike the HP48 choose box engine. The OPENFEM browser is much faster though.

Level 1	→	Level 1
	→	

Remarks:

If the current structure is not "empty" you will be prompted to save the current structure before it is replaced by the opened one. OPENFEM looks for files with the FEM extension (i.e. Truss.FEM) in the current directory.

4.3.6 SAVEFEM

Saves the current structure as a file.

Level 1	→	Level 1
	→	

Remarks:

The FEM extension is automatically added by the program. The structure is saved in the current directory as a Library Data object. Illegal names (i.e. Truss Big) are allowed. Memory can be saved if you have the BZ compressor present on your HP48. Saved files will be BZ compressed first if the BZ option (with BZ¿ (4.3.72) command) is toggled on. An additional saved file size reduction can be accomplishes by purging the calculation results with the PGRS (4.3.12) command before saving the file. Note that then you will have to calculate the structure again after re-opening the file.

4.3.7 SINFO

Displays information about the current structure.

Level 1	→	Level 1
	→	

Remarks:

Also displays erroneous or undefined (but necessary) input.

4.3.8 →FEM

Stores a FEM file as the current structure.

Level 1	→	**Level 1**	
Library Data	→		overwrite current structure
0	→		erase current structure

Remarks:
Intended for programmable use. Overwrites or erases current structure. Checks for validity of the Library Data.

4.3.9 FEM→

Recalls the current structure as a Library Data object.

Level 1	→	**Level 1**
	→	Library Data

Remarks:
Intended for programmable use.

4.3.10 →SNAM

Stores a name string as the name of the current structure.

Level 1	→	**Level 1**
"string"	→	

Remarks:
Intended for programmable use.

4.3.11 SNAM→

Recall the name string of the current structure.

Level 1	→	**Level 1**
	→	"string"

Remarks:
Intended for programmable use.

4.3.12 PGRS

Purges the calculation results of the current structure.

Level 1	→	**Level 1**
	→	

Remarks:
Can be used to save memory when files are saved, see SAVEFEM (4.3.6) command. This command works without confirmation. However, the command asks for confirmation when executed from the FEM (4.3.1) menu. The programmable version of the command is left-shifted available in the FEM menu.

4.3.13 NODE

Starts inputline prompt for nodal X and Z coordinates.

Level 1	→	**Level 1**
	→	

Remarks:
The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet.

4.3.14 →NODE

Stores the nodal X and Z coordinates matrix.

Level 1	→	Level 1	
[[matrix]] $_{n\,x\,2}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the X coordinates and column 2 contains the Z coordinates, see also 2.1. Row 1 equals node 1, row 2 equals node 2 etc.

4.3.15 NODE→

Recalls the nodal X and Z coordinates matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n\,x\,2}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.14.

4.3.16 PROP

Starts inputline prompt for member properties.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>.

4.3.17 →PROP

Stores the member properties matrix.

Level 1	→	Level 1	
[[matrix]] $_{n\,x\,3}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the cross-sectional Area, column 2 contains the moment of inertia Iy and column 3 contains the modulus of elasticity E. Row 1 equals property 1, row 2 equals property 2 etc.

4.3.18 PROP→

Recalls member properties matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n\,x\,3}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.17.

4.3.19 MEMB

Starts inputline prompt for member incidences and the member property reference number.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet.

4.3.20 →MEMB

Stores the member matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 3}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the start node Ni, column 2 contains the end node Nj and column 3 contains the property reference. Row 1 equals member 1, row 2 equals member 2 etc.

4.3.21 MEMB→

Recalls the member matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 3}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.20.

4.3.22 SUPP

Starts inputline prompt for the supported nodes and their restraints.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet.
A value of 0 for a degree of freedom means that movement is free, a value of 1 means that movement is restrained. Spring supports can be assigned by entering a negative number. The absolute value of the number is the stiffness of the spring. Rotational springs are entered in "moment" per radian.

4.3.23 →SUPP

Stores the supports matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 4}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the restrained node Ni, column 2 contains the UX restraint, column 3 contains the UZ restraint and column 4 contains the RY restraint. Row 1 equals support 1, row 2 equals support 2 etc. See also 4.3.22.

4.3.24 SUPP→

Recalls the support matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 4}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.22 and 4.3.23.

4.3.25 MREL

Starts inputline prompt for member end releases data.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>.
As opposed to early versions of FEM48, members with end releases can now be loaded with any type of load. However, please note that the QUERY module is not able to calculate transverse displacements and rotation of members that have a release at the start node Ni. Therefore, if a member needs only one release, it is preferred to orientate the member (choice of Ni and Nj) so that the release is at the end node Nj.
Be aware that a node must be attached to at least one member for rotation, otherwise the structure cannot be solved (the node could then rotate unhindered, this could be solved by giving it a restraint against rotation[6] but it is preferred to leave it attached to a member).

4.3.26 →MREL

Stores the member end releases matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 3}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the member with end release(s), column 2 contains the state of the start release at Ni (0=no release, equals fixed to node Ni and 1=release, equals pinned to node Ni) and column 3 contains the state of the end release at Nj (0=no release, equals fixed to node Nj and 1=release, equals pinned to node Nj). If the member has releases at both the beginning and the end it is turned into a truss element. Row 1 equals member with releases 1, row 2 equals member with releases 2 etc. See also 4.3.25.

4.3.27 MREL→

Recalls the member end releases matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 3}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

[6] Inside information: This is actually the way truss structures are solved. Truss structures use a different local stiffness matrix (only axial stiffness) so rotation of nodes is initially not restrained. This is internally solved by giving each node a restraint against rotation. When viewing the nodal displacements of a truss structure you will notice that the nodes have not rotated.

Remarks:

See 4.3.25 and 4.3.26.

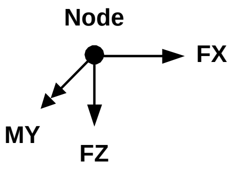
4.3.28 NLF

Starts inputline prompt for nodal force loads.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>.
Nodal force loads are entered in the global coordinates system. Multiple occurrences of a node are allowed.
Below the positive directions are drawn:



4.3.29 →NLF

Stores the nodal force loads matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 4}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded node, column 2 contains the FX load, column 3 contains the FZ load and column 4 contains the MY load. Row 1 equals nodal force load 1, row 2 equals nodal force load 2 etc. See also 4.3.28.

4.3.30 NLF→

Recalls the nodal force loads matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 4}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.28 and 4.3.29.

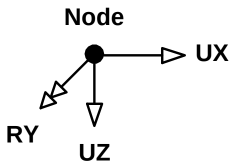
4.3.31 NLD

Starts inputline prompt for nodal displacement loads, where rotation is entered in radians.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>.
Nodal displacement loads are entered in the global coordinates system. Multiple occurrences of a node are allowed, but only if the loaded degree of freedom is different for each occurrence of that node. NLD overrules defined supports. So, to "sink" a support, just load it with a NLD. Unlike with FEM48 version 4.2, there is no need to "unrestrain" the degree of freedom that is loaded with the NLD. Below the positive directions are drawn:



4.3.32 →NLD

Stores the nodal displacement loads matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 4}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded node, column 2 contains the UX load, column 3 contains the UZ load and column 4 contains the RY load (in radians). Row 1 equals nodal displacement load 1, row 2 equals nodal displacement load 2 etc. See also 4.3.31.

4.3.33 NLD→

Recalls the nodal displacement loads matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 4}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.31 and 4.3.32.

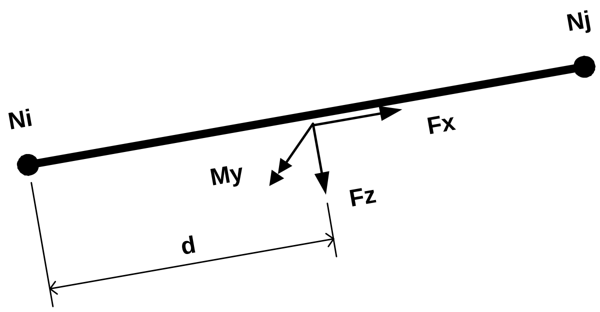
4.3.34 MLC

Starts inputline prompt for concentrated member loads (local axes).

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet.
Member loads are entered in the local coordinates system. Multiple occurrences of a member are allowed.
Below the positive directions are drawn:



4.3.35 →MLC

Stores the concentrated member loads matrix (local axes).

Level 1	→	Level 1	
[[matrix]] $_{n \times 5}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded member, column 2 contains the Fx load, column 3 contains the Fz load, column 4 contains the My load and column 5 contains the distance of the load(s) from the start node Ni. Row 1 equals concentrated member load 1, row 2 equals concentrated member load 2 etc. See also 4.3.34 and 6.3.11.

4.3.36 MLC→

Recalls the concentrated member loads matrix (local axes).

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 5}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.34, 4.3.35 and 6.3.11.

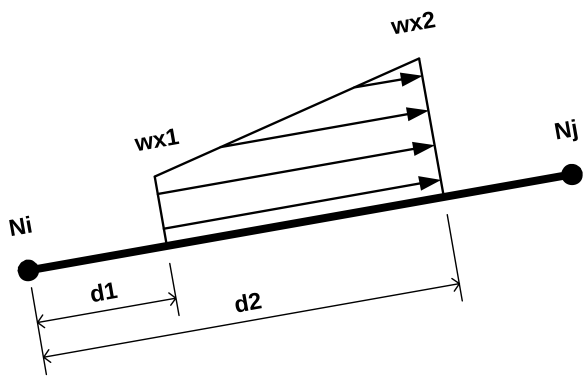
4.3.37 MLX

Starts inputline prompt for trapezoidal distributed axial (local x direction) member loads.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet.
Member loads are entered in the local coordinates system. Multiple occurrences of a member are allowed.
Below the positive directions are drawn:



4.3.38 →MLX

Stores the trapezoidal distributed axial (local x direction) member loads matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 5}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded member, column 2 contains the start load value wx1, column 3 contains the end load value wx2, column 4 contains the distance of the start of the load d1 from the start node Ni and column 5 contains distance of the end of the load d2 from the start node Ni. Note that when the load ends at the end node Nj you can also enter the value 0 for d2. This to avoid calculating the length of inclined members. Row 1 equals trapezoidal distributed axial member load 1, row 2 equals trapezoidal distributed axial member load 2 etc. See also 4.3.37 and 6.3.12.

4.3.39 MLX→

Recalls the trapezoidal distributed axial (local x direction) member loads matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 5}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.37, 4.3.38 and 6.3.12.

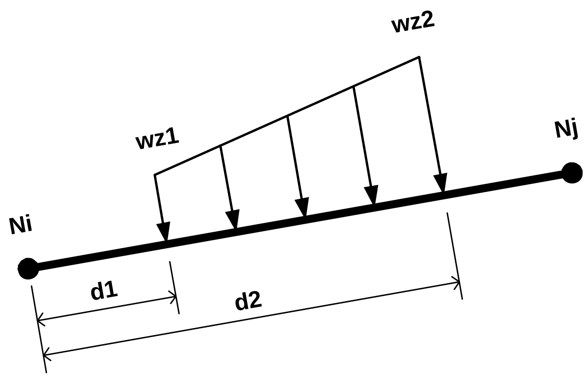
4.3.40 MLZ

Starts inputline prompt for trapezoidal distributed transverse (local z direction) member loads.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>.
Member loads are entered in the local coordinates system. Multiple occurrences of a member are allowed.
Below the positive directions are drawn:



4.3.41 →MLZ

Stores the trapezoidal distributed transverse (local z direction) member loads matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 5}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded member, column 2 contains the start load value wz1, column 3 contains the end load value wz2, column 4 contains the distance of the start of the load d1 from the start node Ni and column 5 contains distance of the end of the load d2 from the start node Ni. Note that when the load ends at the end node Nj you can also enter the value 0 for d2. This to avoid calculating the length of inclined members. Row 1 equals trapezoidal distributed transverse member load 1, row 2 equals trapezoidal distributed transverse member load 2 etc. See also 4.3.40 and 6.3.13.

4.3.42 MLZ→

Recalls the trapezoidal distributed transverse (local z direction) member loads matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 5}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.40, 4.3.41 and 6.3.13.

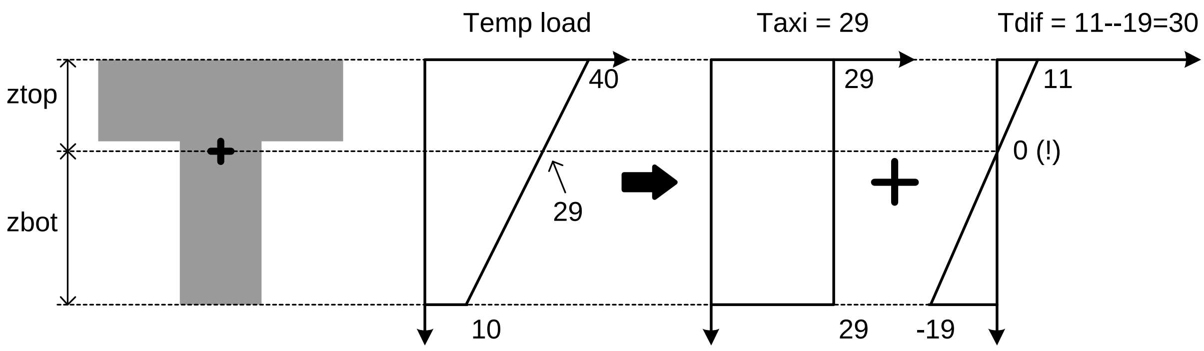
4.3.43 MLT

Starts inputline prompt for temperature member loads.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input <u>if no data line has been [ENTER]ed yet</u>. Multiple occurrences of a member are allowed. See also 10.4 and 10.5. Below the input is explained with a drawing:



Taxi: axial strain / normal stress

Tdif: curvature / bending stress

4.3.44 →MLT

Stores the temperature member loads matrix.

Level 1	→	Level 1	
[[matrix]] $_{n \times 5}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the loaded member (Memb), column 2 contains the axial temperature change (Taxi), column 3 contains the temperature difference between top and bottom (Tdif), column 4 contains the height of the member (h) and column 5 contains the coefficient of thermal expansion (α).
Row 1 equals temperature member load 1, row 2 equals temperature member load 2 etc. See 4.3.43, 10.4 and 10.5.

4.3.45 MLT→

Recalls the temperature member loads matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 5}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 4.3.43 and 4.3.44.

4.3.46 PGLD

Purges all loads on the structure.

Level 1	→	Level 1
	→	

Remarks:

This command works <u>without</u> confirmation. However, the command asks for confirmation when executed from the FEM (4.3.1) menu. The programmable version of the command is left-shifted available in the FEM menu.

4.3.47 SCALC

Calculates the current structure.

Level 1	→	Level 1
	→	

Remarks:

Will execute +/- 10% faster if the FAST¿ (4.3.56) option is enabled. SCALC uses a Cholesky solver instead of the internal HP solver. The Cholesky solver is optimised for speed and memory usage. Exits with SINFO (4.3.7) if input contains errors or when not enough input is present.

4.3.48 SPLOT

Plots the current structure.

Level 1	→	Level 1	
	→		→STK¿ (4.3.70) off
	→	grob	→STK¿ on

Remarks:

The resulting plot depends upon the state of the plot options: DFOR¿ (4.3.62), MAGN (4.3.80), SUPP¿ (4.3.64), NN¿ (4.3.66) and MN¿ (4.3.68).

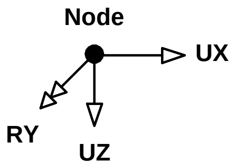
4.3.49 NDIS

Recalls the nodal displacements matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 4}$	AV¿ (4.3.58) off
	→	unknown	AV¿ on

Remarks:

Column 1 contains the node number, column 2 contains the displacement in global x direction, column 3 contains the displacement in global z direction and column 4 contains the rotation around the global y axis (in radians).
Below the positive directions are drawn:



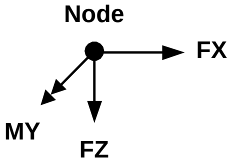
4.3.50 REAC

Recalls the support reactions matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 4}$	AV¿ (4.3.58) off
	→	unknown	AV¿ on

Remarks:

Column 1 contains the node number, column 2 contains the reaction force FX in global x direction, column 3 contains the reaction force FZ in global z direction and column 4 contains the reaction moment MY around the global y axis.
Below the positive directions are drawn:



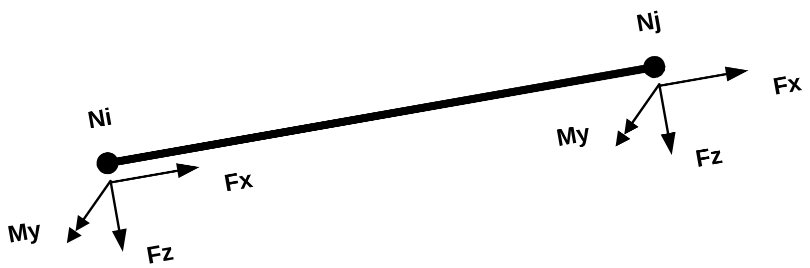
4.3.51 MFOR

Recalls the member end forces matrix.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 5}$	AV¿ (4.3.58) off
	→	unknown	AV¿ on

Remarks:

Column 1 contains the member number, column 2 contains the node number where the end forces are situated, column 3 contains the member end force FX in local x direction, column 4 contains the member end force FX in local z direction and column 5 contains the member end moment MY around the local y axis.
Below the positive directions are drawn:



4.3.52 FRAM¿

Toggles the current structure type.

Level 1	→	Level 1	
1	→		frame structure
0	→		truss structure

Remarks:

The default state of a new structure is frame. When the structure type is set to truss, members have only axial stiffness! Have a look at the MREL command (4.3.25) on how to turn just a few members into truss elements.

4.3.53 FRAM?

Recalls the current structure type.

Level 1	→	Level 1	
	→	1	frame structure
	→	0	truss structure

Remarks:

See 4.3.52.

4.3.54 LCAS¿

Toggles LCASE module mode for FEM command.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:

When enabled <u>and</u> the LCASE (9) module is present, the FEM (4.3.1) command calls the FEM2 (9.4.2) command from the LCASE module, otherwise the standard FEM menu is displayed. When toggled directly from the FEM (or FEM2) menu the FEM2 (or FEM) menu is reloaded, so you will be in the correct menu at once.

4.3.55 LCAS?

Recalls LCASE module mode for FEM command.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

Remarks:

See 4.3.54.

4.3.56 FAST¿

Toggles fast mode.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:

Enabling fast mode turns of the display during calculations which reduces calculation time with +/- 10%.

4.3.57 FAST?

Recalls fast mode.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

Remarks:

See 4.3.56.

4.3.58 AV¿

Toggles auto-viewing of arrays and strings and also toggles trace mode for QUERY plots.

Level 1	→	**Level 1**	
1	→		enabled
0	→		disabled

Remarks:

Matrices are viewed with the viewer defined with the MATV command (4.3.60), strings (PRINT module) with the viewer defined with the STRV command (7.3.9). See 5.3 for information about the trace mode.

4.3.59 AV?

Recalls auto-viewing.

Level 1	→	**Level 1**	
	→	1	enabled
	→	0	disabled

Remarks:

See 4.3.58.

4.3.60 RND¿

Toggles auto-rounding of calculation results.

Level 1	→	**Level 1**	
1	→		enabled
0	→		disabled

Remarks:

The results are saved in full precision. The rounding takes place when they are recalled. Results are rounded with the value stored with the RVAL command, see 4.3.77.

4.3.61 RND?

Recalls auto-rounding of calculation results.

Level 1	→	**Level 1**	
	→	1	enabled
	→	0	disabled

Remarks:

See 4.3.60.

4.3.62 DFOR¿

Toggles deformation plot.

Level 1	→	**Level 1**	
1	→		enabled
0	→		disabled

Remarks:

When toggled on the deformed structure is plotted on top of the undeformed structure. The deformation scale can be set with the MAGN command (4.3.80). Only nodal displacements are taken into account. Member deformations are not plotted, use the QUERY module (5) for this.

4.3.63 DFOR?

Recalls deformation plot.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

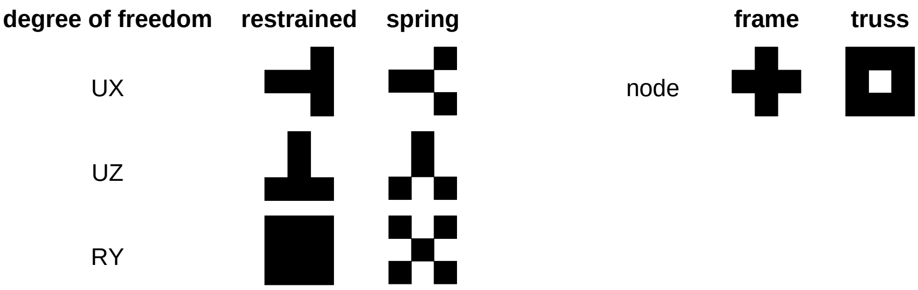
Remarks:
See 4.3.62.

4.3.64 SUPP¿

Toggles the plotting of supports

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:
Supports can not be plotted at the same time as the node numbers, see NN¿ command (4.3.66).



4.3.65 SUPP?

Recalls the plotting of supports

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

Remarks:
See 4.3.64.

4.3.66 NN¿

Toggles the plotting of node numbers.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:
Node numbers can not be plotted at the same time as the supports, see SUPP¿ command (4.3.64).

4.3.67 NN?

Recalls the plotting of node numbers.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

Remarks:

See 4.3.66.

4.3.68 MN¿

Toggles the plotting of member numbers.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

4.3.69 MN?

Recalls the plotting of member numbers.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

4.3.70 →STK¿

Toggles the export of plots to the stack.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

4.3.71 →STK?

Recalls the export of plots to the stack.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

4.3.72 BZ¿

Toggles BZ compression of files while saving them.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:

See 4.3.6. The BZ compressor should be available as a library command. A program in a directory is not supported.

4.3.73 BZ?

Recalls BZ compression of files while saving them.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

Remarks:
See 4.3.72.

4.3.74 MATV

Starts inputline prompt for an external matrixwriter/viewer.

Level 1	→	Level 1
	→	

Remarks:
External matrixwriter/viewer, can also be a text editor. Command should be a library command, a variable name is not allowed. When no external matrixwriter/viewer is defined the internal HP48 matrixwriter is used. Also see 4.3.58.

4.3.75 →MATV

Stores the external matrixwriter/viewer command as a string.

Level 1	→	Level 1
"string"	→	

Remarks:
A nullstring sets the use of the internal matrixwriter. See also 4.3.74.

4.3.76 MATV→

Recalls the external matrixwriter/viewer command as a string.

Level 1	→	Level 1
	→	"string"

Remarks:
See also 4.3.74.

4.3.77 RVAL

Starts inputline prompt for the rounding value.

Level 1	→	Level 1
	→	

Remarks:
The rounding follows the conventions of the HP RND command:

$0 \leq n \leq 11$	round to n decimal places
$-11 \leq n \leq -1$	round to n significant digits
$n = 12$	round to current display format

4.3.78 →RVAL

Stores the rounding value.

Level 1	→	Level 1
integer	→	

Remarks:
See 4.3.77.

4.3.79 RVAL→

Recalls the rounding value.

Level 1	→	Level 1
	→	integer

Remarks:

See 4.3.77.

4.3.80 MAGN

Starts inputline prompt for the deformation magnification.

Level 1	→	Level 1
	→	

Remarks:

A value of 1000 yields the plotting of 1 millimetre as 1 meter. The deformation magnification must be greater than zero.

4.3.81 →MAGN

Stores the deformation magnification.

Level 1	→	Level 1
real number	→	

Remarks:

See 4.3.80.

4.3.82 MAGN→

Recalls the deformation magnification.

Level 1	→	Level 1
	→	real number

Remarks:

See 4.3.80.

4.3.83 FBROW

Gives the user access to the custom browser used by FEM48 (much faster than the HP48 choose engine).

Level 4	Level 3	Level 2	Level 1	→	Level 2	Level 1	
"string1"	"string2"	{ choices }	init_pos	→	choice	choice_pos	ENTER key
"string1"	"string2"	{ choices }	init_pos	→		0	ON key

Remarks:

For use by user's own UserRPL programs. Is fastest when used with a choice list comprising of strings, since HP48 internal string conversion routine can be very slow for large objects (related to stack speed!). Only available from the FEM48 library menu, since unrelated to FEM48 workings.

4.3.84 ABOUTFEM

Recalls the version string.

Level 1	→	Level 1
	→	"string"

5. QUERY MODULE

5.1 Description

This module is optional and provides beam analysis facilities. The member(s) to be analysed can be selected after which numerical, tabular and graphical output of;

- axial force
- shear force
- bending moment
- rotation
- axial displacements
- transverse displacements
- shear stress
- combined axial and bending stress at top/bottom
- Huber-Hencky stress at top/bottom

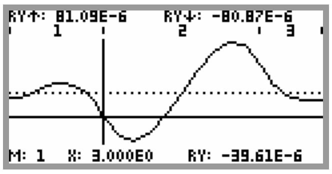
can be obtained.

5.2 Menu structure

See 1.5 for a complete menu description of all FEM48 modules.

5.3 Trace mode

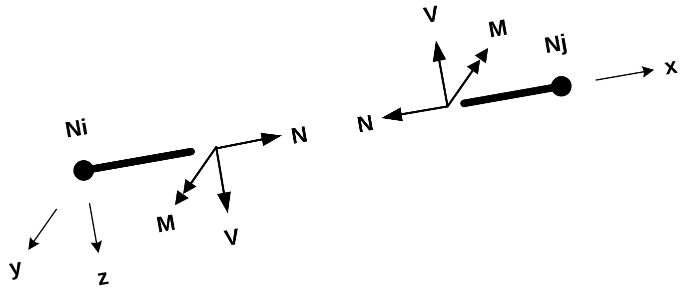
The query plots have a trace mode, that is activated when the AV¿ (4.3.58) option is enabled. For the current keypoint the member number, the local x value and the N, V, M, UX, RY or UZ value is displayed below the plot. Maxima are displayed above the plot as usual. See the screenshot below for an impression.

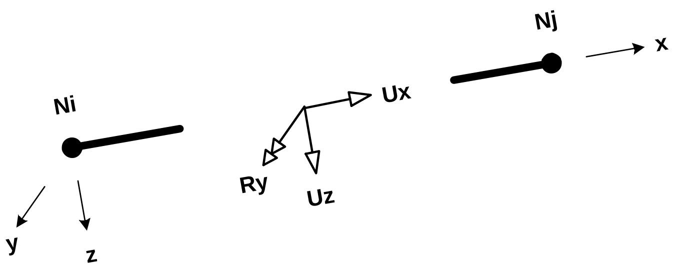


The following keys are active during trace mode:

key	plane	action
[←]	unshifted	go left to next keypoint (with keyrepeat)
[←]	right-shifted	go all left
[→]	unshifted	go right to next keypoint (with keyrepeat)
[→]	right-shifted	go all right
[↑]	unshifted	go right to next member (with keyrepeat)
[↓]	unshifted	go left to next member (with keyrepeat)
[A]	unshifted	go all left
[B]	unshifted	go all right
[ENTER]	unshifted	copy current keypoint to stack, format depends on TAG¿ (5.6.28)
[ON]	unshifted	quit trace mode, return to stack
[ON]	right-shifted	turn calculator off

5.4 Sign of internal forces and displacements





5.5 Stress formulas

The following formulas are used to calculate stresses, using the PROP and PROP2 property definitions.

$$\tau X = \frac{VX}{Area_{PROP2}} \qquad \sigma TX = \frac{NX}{Area_{PROP}} - \frac{MX \cdot Ztop}{Iy} \qquad \sigma BX = \frac{NX}{Area_{PROP}} + \frac{MX \cdot Zbot}{Iy}$$

$$\sigma iTX = \sqrt{\sigma TX^2 + 3 \cdot \tau X^2} \qquad \sigma iBX = \sqrt{\sigma BX^2 + 3 \cdot \tau X^2}$$

5.6 Command reference

5.6.1 ABOUTFEMQUERY

Recalls the version string.

Level 1	→	Level 1
	→	"string"

5.6.2 QUERYFEM

Displays the QUERY menu.

Level 1	→	Level 1
	→	

Remarks:

Normally accessed through the use of the FEM menu (4.3.1).

5.6.3 MINFO

Displays information about the selected member.

Level 1	→	Level 1
integer	→	

5.6.4 KEYP

Starts inputline prompt for the number of keypoints along the queried member.

Level 1	→	Level 1
	→	

Remarks:

Member query data will only be calculated at the keypoints. Number of keypoints allowed: $2 \leq$ keypoints ≤ 100. Keypoints are evenly distributed over the member.

5.6.5 →KEYP

Stores the number of keypoints along the queried member.

Level 1	→	Level 1
integer	→	

Remarks:

See 5.6.4.

5.6.6 KEYP→

Recalls the number of keypoints along the queried member.

Level 1	→	Level 1
	→	integer

Remarks:

See 5.6.4.

5.6.7 NX

Calculates the axial force at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.8 VX

Calculates the shear force at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.9 MX

Calculates the bending moment at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:
Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.10 UXX

Calculates the axial displacement at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:
Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.11 RYX

Calculates the rotation in radians at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:
Level 2: member number, level 1: local x coordinate. Calculation errors if queried member has a release at the start node Ni. See also 5.6.29.

5.6.12 UZX

Calculates the transverse displacement at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:
Level 2: member number, level 1: local x coordinate. Calculation errors if queried member has a release at the start node Ni. See also 5.6.29.

5.6.13 NTAB

Creates a table of values for the axial force at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:
Column 1 contains the local x coordinates and column 2 contains the axial force. See also 5.6.4 and 5.6.29.

5.6.14 VTAB

Creates a table of values for the shear force at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:
Column 1 contains the local x coordinates and column 2 contains the shear force. See also 5.6.4 and 5.6.29.

5.6.15 MTAB

Creates a table of values for the bending moment at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the bending moment. See also 5.6.4 and 5.6.29.

5.6.16 UXTAB

Creates a table of values for the axial displacement at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the axial displacement. See also 5.6.4 and 5.6.29.

5.6.17 RYTAB

Creates a table of values for the rotation in radians at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the rotation. Calculation errors if queried member has a release at the start node Ni. See also 5.6.4 and 5.6.29.

5.6.18 UZTAB

Creates a table of values for the transverse displacement at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the transverse displacement. Calculation errors if queried member has a release at the start node Ni. See also 5.6.4 and 5.6.29.

5.6.19 NPLT

Plots the axial force of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.20 VPLT

Plots the shear force of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.21 MPLT

Plots the bending moment of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. The positive direction (upwards or downwards) depends on the state of ↑M? (5.6.27). An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.22 UXPLT

Plots the axial displacement of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.23 RYPLT

Plots the rotation (in radians) of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. Errors if queried member has a release at the start node Ni. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.24 UZPLT

Plots the transverse displacement of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:
The queried member, the minimum and the maximum value are displayed above the plot. Errors if queried member has a release at the start node Ni. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.25 AXIS¿

Toggles the plotting of the x axis.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

5.6.26 AXIS?

Recalls the plotting of the x axis.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

5.6.27 ↑M¿

Toggles positive direction of bending moment plot.

Level 1	→	Level 1	
1	→		positive upwards
0	→		positive downwards

5.6.28 ↑M?

Recalls positive direction of bending moment plot.

Level 1	→	Level 1	
	→	1	positive upwards
	→	0	positive downwards

5.6.29 TAG¿

Toggles the tagging of stack results and exported plots.

Level 1	→	Level 1	
1	→		enabled
0	→		disabled

Remarks:
Note that matrix output is NOT tagged when the AV¿ option is enabled (4.3.58).

5.6.30 TAG?

Recalls the tagging of stack results and exported plots.

Level 1	→	Level 1	
	→	1	enabled
	→	0	disabled

5.6.31 PROP2

Starts inputline prompt for additional member properties, needed for stress calculations.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (overwrites) the data type in the current structure or 2) cancels input if no data line has been [ENTER]ed yet. The number of properties of PROP2 need to be the same as that of the PROP (4.3.16) properties definitions.

5.6.32 →PROP2

Stores the additional member properties matrix, needed for stress calculations.

Level 1	→	Level 1	
[[matrix]] $_{n \times 3}$	→		overwrite
0	→		erase

Remarks:

Column 1 contains the cross-sectional Area for shear[7], column 2 contains the distance from the top of the section to the elastic line and column 3 contains the distance from the bottom of the section to the elastic line. Row 1 equals property 1, row 2 equals property 2 etc. See 5.6.31.

5.6.33 PROP2→

Recalls the additional member properties matrix, needed for stress calculations.

Level 1	→	Level 1	
	→	[[matrix]] $_{n \times 3}$	defined and AV¿ (4.3.58) off
	→	unknown	defined and AV¿ on
	→	0	undefined

Remarks:

See 5.6.31.

5.6.34 τX

Calculates the shear stress at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.35 σTX

Calculates the combined axial and bending stress for the top of the section at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

[7] can be different from the cross-sectional area for axial stress (H-beams for instance)

5.6.36 σBX

Calculates the combined axial and bending stress for the bottom of the section at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.37 σiTX

Calculates the Huber-Hencky stress for the top of the section at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.38 σiBX

Calculates the Huber-Hencky stress for the bottom of the section at the local x coordinate of the queried member.

Level 2	Level 1	→	Level 1
integer	real number	→	real number

Remarks:

Level 2: member number, level 1: local x coordinate. See also 5.6.29.

5.6.39 τTAB

Creates a table of values for the shear stress at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the shear stress. See also 5.6.4 and 5.6.29.

5.6.40 σTTAB

Creates a table of values for the combined axial and bending stress for the top of the section at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the combined axial and bending stress for the top of the section. See also 5.6.4 and 5.6.29.

5.6.41 σBTAB

Creates a table of values for the combined axial and bending stress for the bottom of the section at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the combined axial and bending stress for the bottom of the section. See also 5.6.4 and 5.6.29.

5.6.42 σiTTAB

Creates a table of values for the Huber-Hencky stress for the top of the section at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the shear stress. See also 5.6.4 and 5.6.29.

5.6.43 σiBTAB

Creates a table of values for the Huber-Hencky stress for the bottom of the section at the local x coordinate of the queried member at each keypoint.

Level 1	→	Level 1	
integer	→	[[matrix]] $_{n \times 2}$	AV¿ (4.3.58) off
integer	→	unknown	AV¿ on

Remarks:

Column 1 contains the local x coordinates and column 2 contains the shear stress. See also 5.6.4 and 5.6.29.

5.6.44 τPLT

Plots the shear stress of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.45 σTPLT

Plots the combined axial and bending stress for the top of the section of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.46 σBPLT

Plots the combined axial and bending stress for the bottom of the section of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.47 σiTPLT

Plots the Huber-Hencky stress for the top of the section of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

5.6.48 σiBPLT

Plots the Huber-Hencky stress for the bottom of the section of the queried member(s) at each keypoint.

Level 1	→	Level 1	
integer or list of integers	→		→STK¿ (4.3.70) off
integer or list of integers	→	grob	→STK¿ on

Remarks:

The queried member, the minimum and the maximum value are displayed above the plot. An empty list as argument plots all members. Trace mode is activated when the AV¿ (4.3.58) option is enabled, see 5.3. See also 5.6.4 and 5.6.29.

6. WIZRD MODULE

6.1 Description

This module is optional and provides wizards for entering geometry and properties of "standard" structures like beams, bay frames and lattices and for generating loads like selfweight and pre/post-stressing cables. A link to SED48, a Structural Engineering Database by the author of FEM48 is provided.

6.2 Menu structure

See 1.5.

6.3 Command reference

6.3.1 ABOUTFEMWIZRD

Recalls the version string.

Level 1	→	Level 1
	→	"string"

6.3.2 WIZRDFEM

Displays the WIZRD menu.

Level 1	→	Level 1
	→	

Remarks:

Normally accessed through the use of the FEM menu (4.3.1).

6.3.3 RBEAM

Starts inputline prompt for regular beam geometry.

Level 1	→	Level 1
	→	

Remarks:

The length per span and the number of spans are entered after which the nodes, members and supports are generated and stored (overwriting!) in the current structure. Structure type is set to FRAME. All other data is untouched.
All spans have the same length. All members have property reference 1. Each node is supported with a roll, with the exception of the first node, this has a pinned support.

6.3.4 IBEAM

Starts inputline prompt for irregular beam geometry.

Level 1	→	Level 1
	→	

Remarks:

The length per span is entered after which the nodes, members and supports are generated and stored (overwriting!) in the current structure. Structure type is set to FRAME. All other data is untouched. All members have property reference 1. Each node is supported with a roll, with the exception of the first node, this has a pinned support.

6.3.5 BAYFR

Starts inputline prompt for bayframe geometry.

Level 1	→	Level 1
	→	

Remarks:

The width per bay, the height per bay, the number of horizontal bays and the number of vertical bays are entered after which the nodes, members and supports are generated and stored (overwriting!) in the current structure. Structure type is set to FRAME. All other data is untouched. Horizontal members have property reference 1, vertical members have property reference 2. The nodes on the bottom row have a pinned support.

6.3.6 LATTI

Starts inputline prompt for lattice geometry.

Level 1	→	Level 1
	→	

Remarks:

The length per bay, the height and the number of bays are entered after which the nodes, members and supports are generated and stored (overwriting!) in the current structure. Structure type is set to TRUSS. All other data is untouched. Horizontal members have property reference 1, vertical members have property reference 2 and diagonal members have property reference 3. The leftmost node on the bottom row has a pinned support, the rightmost node on the bottom row has a rolling support.

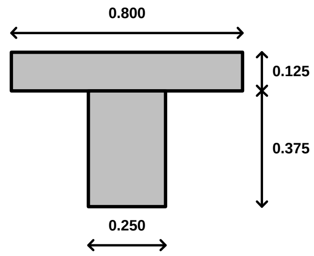
6.3.7 GPROP

Starts properties generation browser.

Level 1	→	Level 1
	→	

Remarks:

A cross section type can be selected: Rectangle (multiple), Circle, Circular Ring, SED48 database, Numerical, and generated. The first cross section generated has property reference 1, the second one property reference 2, etc. The ON key quits the generation after which the properties are stored (overwriting!) in the current structure. All other data is untouched.



Rectangle (multiple)

This wizard can be used to generate properties of sections that are made of multiple rectangular sections, like T and I sections (and normal rectangles of course).
For example, a T section (see picture) with a modulus of elasticity of 3E7 is entered like this: `[ 0.8 0.125 0.25 0.375 3E7 ]`
After the calculation the properties h, A, e, I and E are displayed on the screen and the program halts until a key is pressed[8]. When the [UpArrow] key is pressed the properties are also exported to the stack, for further calculations by the user.

SED48 Database link

The database link offers selection of a profile in the current SED48 database. You can select either Ix, Iy or Iz as moment of inertia[9]. The link then searches the database for the properties { “A“ “Ix“ “E“ }, { “A“ “Iy” “E“ } or { “A“ “Iz“ “E“ } of the selected profile. Since FEM48 and SED48 both don't support units there is a unit conversion routine available for SED48 to FEM48.

6.3.8 SELFW

Starts inputline prompt for the generation of selfweight.

[8] Only when a multiple rectangular section has been entered.
[9] Ix is offered here for users who use different axes for their section properties. For the FEM48 axes this would be the moment of inertia for torsion. Note that to select Ix you have to scroll up in the browser!

Level 1	→	Level 1
	→	

Remarks:

Selfweight can be generated for arbitrary inclined members. The density for each property reference has to be entered. Since each property has defined the cross sectional area the selfweight can be calculated (density * area). The selfweight is then stored in the current structure as MLX and/or MLZ loads. All other data is untouched. Note that the selfweight is added to the structure, no loads are replaced or overwritten. Note that it is thus possible to load a structure with it's selfweight more than once!

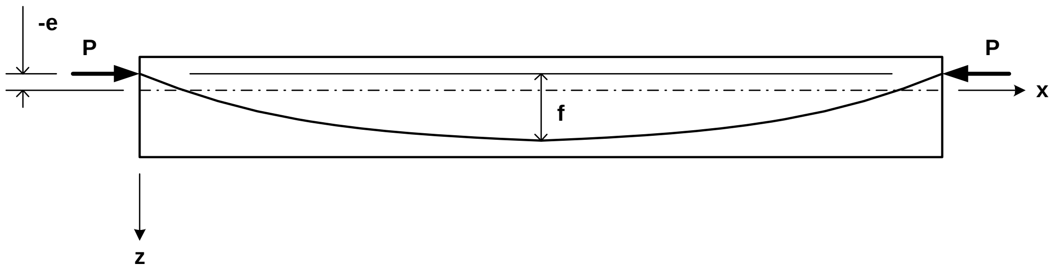
6.3.9 POCAB

Starts inputline prompt for the generation of forces introduced by a parabolic post-stressing cable.

Level 1	→	Level 1
	→	

Remarks:

The member, post-stressing force and the geometrical data of the cable must be entered. After this the loads introduced by the cable are calculated and added to the structure (no loads are replaced or overwritten). Note that by entering f=0 a straight cable can be generated.



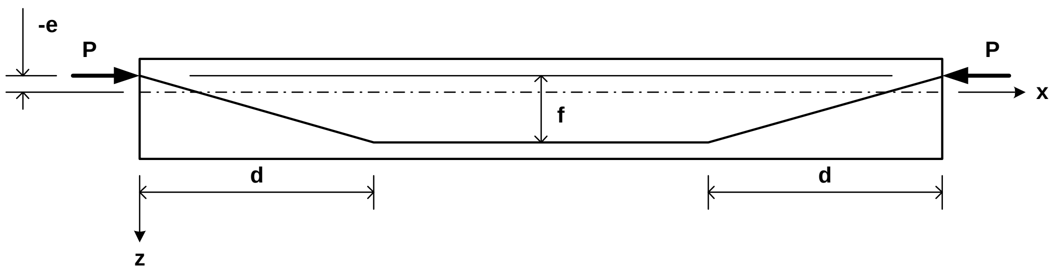
6.3.10 PRCAB

Starts inputline prompt for the generation of forces introduced by a polygon pre-stressing cable.

Level 1	→	Level 1
	→	

Remarks:

The member, pre-stressing force and the geometrical data of the cable must be entered. After this the loads introduced by the cable are calculated and added to the structure (no loads are replaced or overwritten). Note that by entering f=0 a straight cable can be generated.



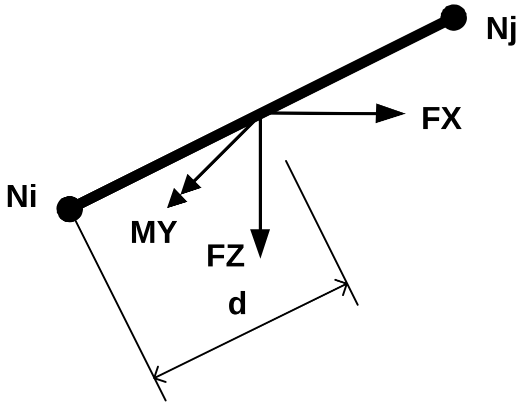
6.3.11 MLCG

Starts inputline prompt for the generation of MLC loads (concentrated member loads) for the global axes system.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (adds, no loads are replaced or overwritten) the data to the MLC loads in the current structure with the transformation from global to local axes system or 2) cancels input if no data line has been [ENTER]ed yet.
Member loads are entered in the global coordinates system. Multiple occurrences of a member are allowed. See also 4.3.34. Below the positive directions are drawn:



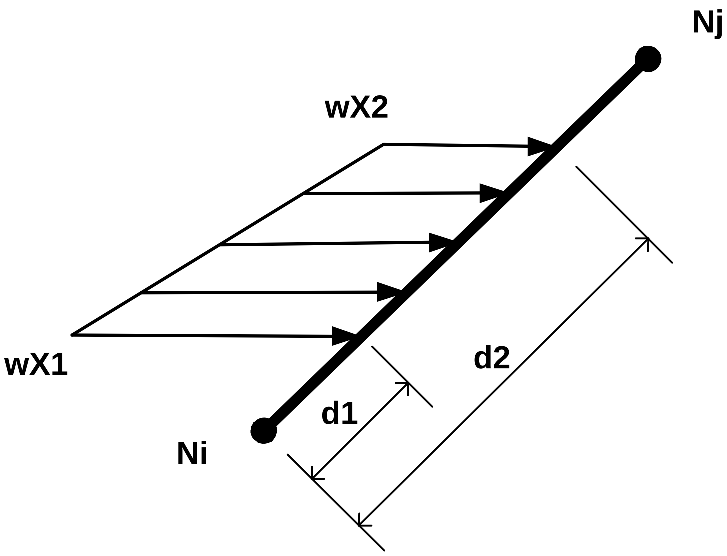
6.3.12 MLXG

Starts inputline prompt for the generation of MLX loads (trapezoidal distributed member loads) for the global axes system.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (adds, no loads are replaced or overwritten) the data to the MLX and MLZ loads in the current structure with the transformation from global to local axes system or 2) cancels input if no data line has been [ENTER]ed yet.
Member loads are entered in the global coordinates system. Multiple occurrences of a member are allowed. See also 4.3.37 and 4.3.40. Below the positive directions are drawn:



6.3.13 MLZG

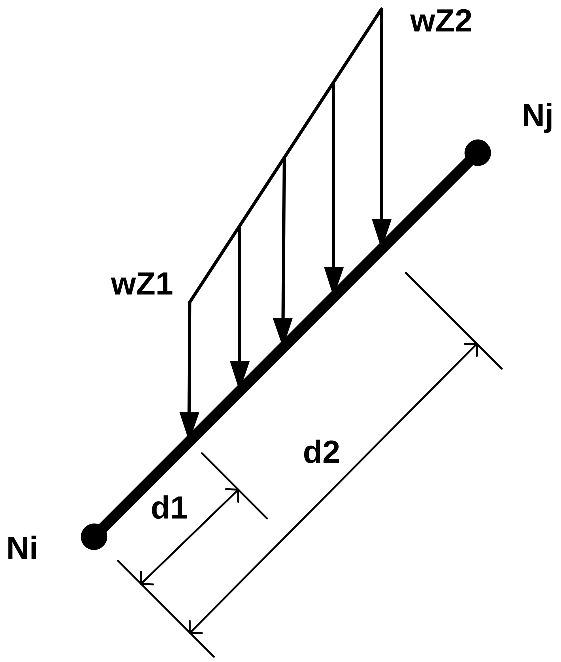
Starts inputline prompt for the generation of MLZ loads (trapezoidal distributed member loads) for the global axes system.

Level 1	→	Level 1
	→	

Remarks:

The [ENTER] key confirms data line and proceeds to next data line, the [ON] key signals the end of input and 1) saves (adds, no loads are replaced or overwritten) the data to the MLX and MLZ loads in the current structure with the transformation from global to local axes system or 2) cancels input if no data line has been [ENTER]ed yet.

Member loads are entered in the global coordinates system. Multiple occurrences of a member are allowed. See also 4.3.37 and 4.3.40. Below the positive directions are drawn:



7. PRINT MODULE

7.1 Description

This module is optional and provides commands for easy viewing and printing[10] of the input and output of FEM48.

7.2 Menu structure

See 1.5 for a complete menu description of all FEM48 modules.

7.3 Command reference

7.3.1 ABOUTFEMPRINT

Recalls the version string.

Level 1	→	Level 1
	→	"string"

7.3.2 PRINTFEM

Displays the PRINT menu.

Level 1	→	Level 1
	→	

Remarks:

Normally accessed through the use of the FEM menu (4.3.1).

7.3.3 I$

Assembles a pretty print string of the input of the current structure.

Level 1	→	Level 1	
	→	"string"	AV¿ (4.3.58) off
	→	unknown	AV¿ on

7.3.4 O$

Assembles a pretty print string of the calculated results (output) of the current structure.

Level 1	→	Level 1	
	→	"string"	AV¿ (4.3.58) off
	→	unknown	AV¿ on

7.3.5 IO$

Assembles a pretty print string of the input and the calculated results (output) of the current structure.

Level 1	→	Level 1	
	→	"string"	AV¿ (4.3.58) off
	→	unknown	AV¿ on

7.3.6 PI$

Prints a pretty print string of the input of the current structure.

[10] If you own an HP printer. Note that it is also easy when using FEM48 on an emulator, this enables you to copy/paste the strings to a text editor on your PC.

Level 1	→	Level 1
	→	

7.3.7 PO$

Prints a pretty print string of the calculated results (output) of the current structure.

Level 1	→	Level 1
	→	

7.3.8 PIO$

Prints a pretty print string of the input and the calculated results (output) of the current structure.

Level 1	→	Level 1
	→	

7.3.9 STRV

Starts inputline prompt for an external stringviewer.

Level 1	→	Level 1
	→	

Remarks:

Can also be a text editor. Command should be a library command, a variable name is not allowed. When no external viewer is defined the internal HP48 editor is used. Also see 4.3.58.

7.3.10 →STRV

Stores the external stringviewer command as a string.

Level 1	→	Level 1
"string"	→	

Remarks:

A nullstring sets the use of the internal editor. See also 7.3.9.

7.3.11 STRV→

Recalls the external stringviewer command as a string.

Level 1	→	Level 1
	→	"string"

Remarks:

See 7.3.9.

8. MOVLD MODULE

8.1 Description

This module is optional and provides a command for generating moving loads along a selection of (or all) members of the current structure.

- define load system of an unlimited number of concentrated member loads and the interval + stepsize
- outputs the NDIS REAC MFOR minimums and maximums in the current directory (as user variables)
- when defined the user program MOVLD.USR is run after each step, so you can add your own postprocessing

8.2 Menu structure

There are only two commands available, MOVLD and ABOUTFEMMOVLD. The MOVLD command is available (left-shift SCALC or SCALC2) from the FEM (4.3.1) menu. There is no separate MOVLD menu available from the FEM menu. See 1.5 for a complete menu description of all FEM48 modules.

8.3 Command reference

8.3.1 ABOUTFEMMOVLD

Recalls the version string.

Level 1	→	**Level 1**
	→	"string"

8.3.2 MOVLD

Calculates minimum and maximum values for the NDIS, REAC and MFOR output for a given selection of members, a load system and a placement interval and stepsize. A user program (MOVLD.USR) can be run after each iteration to perform any analysis desired. Load system loads are added to the already defined loads.

Level 3	**Level 2**	**Level 1**	→	**Level 1**
{ members }	[[matrix]] $_{n \times 4}$	[vector] $_{1 \times 3}$	→	

Remarks:

Member selection

The member selection consists of a list with the member numbers. If all members need to be selected, an empty list can be entered, this for ease of use. Note that the selected members can have an arbitrary inclination. Also, the structure does not need to be a beam. Thus, the structure could be a complete bridge, where you would move the loads along the bridge deck only.

Load system definition

The load system is defined by a matrix:

$$
\begin{array}{l}
[[\ FX_1\ FZ_1\ MY_1\ x_1\ ]\\
\ [\ FX_i\ FZ_i\ MY_i\ x_i\ ]\\
\ [\ FX_n\ FZ_n\ MY_n\ x_n\ ]]
\end{array}
$$

It is understood that $x_i<x_{i+1}$. The x value of the first load of the load system is the reference point for the load system and should be 0. The x values are local "load system" coordinates but are added to the “global” coordinates of the selected members for positioning.

Interval definition

The interval and stepsize are defined by a vector: $[\ X_{start}\ X_{end}\ X_{step}\ ]$

The X values are defined in the “global” coordinates system of the selected members. For the first iteration the load system will be placed on the “global” X coordinate: $X_{start}+x_1$

This means that a non zero value of x for the first load(s) of the system will lead to a displacement of the load system from the global X_{start} coordinate! Negative X_{start} and X_{end} values are also allowed (this in order to let the load system move completely over the selected members). Interval bounderies can fall beyond the beam at both ends if needed.

Output

After finishing, the minima (most negative) and maxima (most positive) of NDIS, REAC and MFOR are stored in the current directory as matrices (user variables: ↑NDIS, ↓NDIS, ↑REAC, ↓REAC, ↑MFOR and ↓MFOR).

UserRPL program

After each iteration, the program MOVLD.USR (when present in current or a parent directory) is run. This to enable the user to perform a secondary analysis for each step. For user's sake, the following "compiled local variables" are available:

←X : the current “global” X position of the load system on the selected members
←M : the current member loaded by the first load of the load system
←x : the current local x position of the first load of the load system on the member ←M
←L : the total length of the selected members

The MOVLD.USR program should not alter the stack (but is protected in case it does).

Examples:

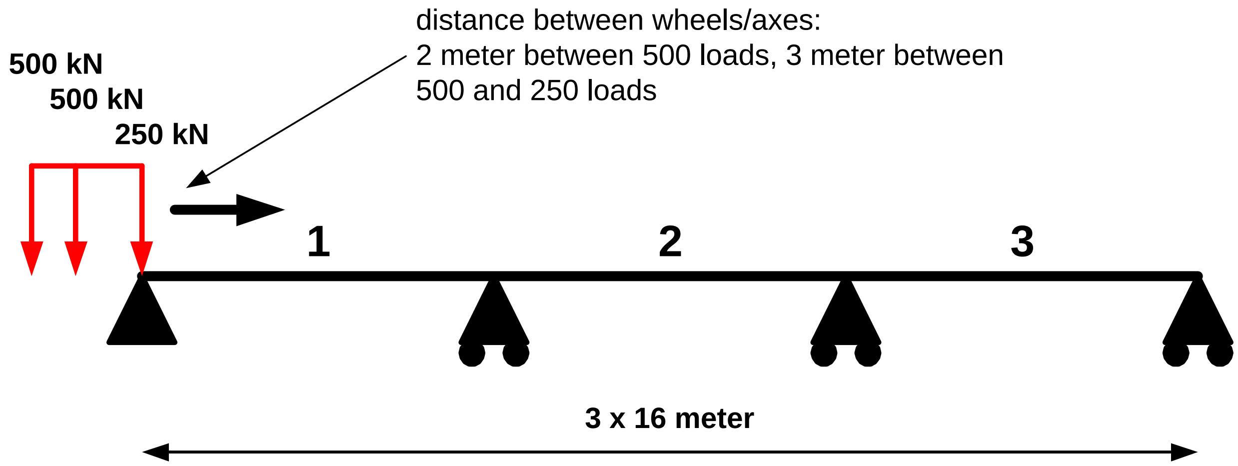
Display the local x value and the current MLC

```
<<
  0 AV¿                      @ no autoview
  "Current local pos: "      @ text string for info
  ←x + 4 DISP                @ add local x pos to string and display it on line 4
  "Current MLC:" 5 DISP      @ text string for info display it on line 5
  MLC→ 6 DISP                @ recall MLC and display it on line 6
>>
```

Influence of moving load on moment at a certain point of a member (using QUERY module and ΣDAT)

```
<<
  0 TAG¿              @ no tags
  ←X                  @ current global X pos, used for identification of load pos
  1 1.5 MX            @ moment of member 1 at local x = 1.5 for current load pos
  2 →ARRY Σ+          @ create vector and sto it in ΣDAT
>>
```

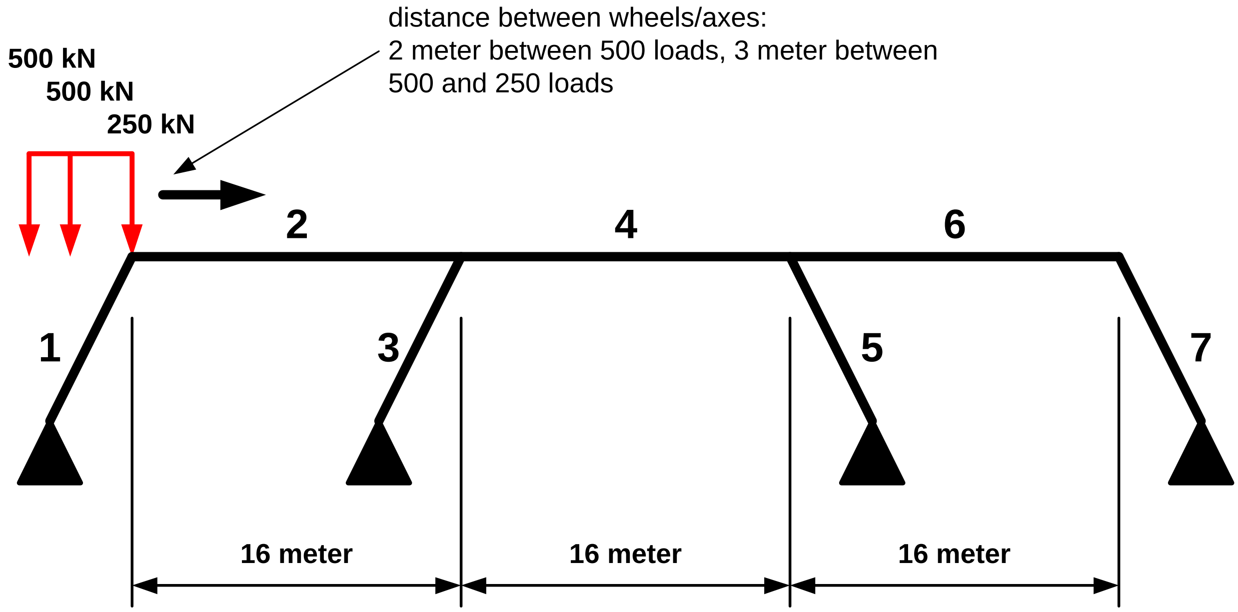
Example 1 of MOVLD use



In example 1, the structure consists of a continious beam. The defined load system will be moved completely along the beam (members 1, 2 and 3). The stepsize is set at 1 meter.

Level 3	Level 2	Level 1	→	Level 1
{ 1 2 3 }	[[0 500 0 0]	[-5 48 1]	→	
or	[0 500 0 2]			
{ }	[0 250 0 5]]			

Example 2 of MOVLD use



In example 2, the structure consists of a frame with a bridge deck. The defined load system will be moved completely along the deck (members 2, 4 and 6). The stepsize is set at 1 meter.

Level 3	Level 2	Level 1	→	Level 1
{ 2 4 6 }	[[0 500 0 0]	[-5 48 1]	→	
	[0 500 0 2]			
	[0 250 0 5]]			

Note that the global coordinates of the nodes have no influence on the “global” X coordinates of the load system interval (level 1 argument). The selected members define a new “global” coordinate system, where X=0 at the start node N_i of the first selected member and $X=L_{total}$ at the end node Nj of the last selected member. L_{total} is the total length of the selected members. There is no check for reversed and/or differently inclined members.

9. LCASE MODULE

9.1 Description

This module is optional and provides commands to manage load cases and load combinations with FEM48.
Features are:

- Define unlimited number of load cases
- Define unlimited number of load combinations
- NLD loads are NOT treated as loads but as part of the geometry definition (supports) and are NOT available for load cases and combinations (one NLD definition for all cases and combinations, just like supports)
- The SCALC2 command calls the SCALC (4.3.47) command for each[11] load case

9.2 Menu structure

The LCASE module provides the FEM2 command, which is akin to the FEM command, but redefines and adds some commands in order to handle load cases and combinations. Most notably the CASEPRE, COMBPRE, SCALC2, CASEPOST and COMBPOST commands are added, see below. See 1.5 for a complete menu description of all FEM48 modules.

9.3 How to use the LCASE module

Below is as short description on how to use the LCASE module, also see the flowchart in section 9.5:

- define your geometry: NODE, MEMB, PROP, SUPP, MREL and NLD (!)
- repeat sequence below for all load cases:
    - o define loads: NLF, MLC, MLX, MLZ and MLT
    - o save loads as a load case with the CASEPRE command (New Case)
- repeat sequence below for all load combinations:
    - o define load combinations with the COMBPRE[12] command (New Comb)
- calculate structure with the SCALC2 command
- choose load case or combination for which to view the results with the CASEPOST or the COMBPOST command
- all other commands of the FEM48, QUERY, PRINT and WIZRD modules work exactly the same as before

9.4 Command reference

9.4.1 ABOUTFEMLCASE

Recalls the version string.

Level 1	→	Level 1
	→	"string"

9.4.2 FEM2

Displays the FEM2 menu.

Level 1	→	Level 1
	→	

Remarks:

See also the LCAS¿ (4.3.54) command.

[11] thus the stiffness matrix is also assembled for each load case (!), alas this must be so because of the modular approach/memory
[12] actually this is also possible after executing SCALC2, since load combinations are assembled “on the fly”

9.4.3 CASEPRE

Load cases pre-processor. With this command you can save current loads as a new load case, overwrite a load case with the current loads, edit a load case, rename a load case or delete a load case within a browser environment.

Level 1	→	Level 1
	→	

Remarks:

- New Case — Saves current loads as a load case with the name provided (after this the loads of load case are erased from NLF, MLC etc)
- Overwrite Case — Overwrite the selected load case with the current loads (after this the loads of load case are erased from NLF, MLC etc)
- Edit Case — Makes the selected load case current (then edit NLF, MLC etc. and Overwrite Case)
- Rename Case — Renames the selected load case
- Delete Case — Deletes selected load case (or all), load case(s) are also deleted from the combinations

9.4.4 COMBPRE

Load combinations pre-processor. With this command you can manage load combinations from within a browser environment.

Level 1	→	Level 1
	→	

Remarks:

- New Combi — Creates a new load combination of the already defined load cases
- Edit Combi — Edits an already defined load combination
- Rename Combi — Renames the selected load combination
- Delete Combi — Deletes the selected load combination (or all)

9.4.5 SCALC2

Calculates the current structure and all defined load cases.

Level 1	→	Level 1
	→	

Remarks:

The geometry defined at the time of the calculation is saved and restored by the CASEPOST and COMBPOST commands. If there is only one load case present, the results of this load case are auto-loaded with CASEPOST. If there are more load cases present you will have to do this yourself with the CASEPOST (9.4.6) and COMBPOST (9.4.7) commands. Will execute +/- 10% faster if the FAST¿ (4.3.56) option is enabled. SCALC2 uses a Cholesky solver instead of the internal HP solver. The Cholesky solver is optimised for speed and memory usage.

9.4.6 CASEPOST

Selects the load case for post-processing.

Level 1	→	Level 1
	→	

9.4.7 COMBPOST

Selects the load combination for post-processing.

Level 1	→	Level 1
	→	

9.4.8 SINFO2

Displays information about the current structure and the current load case / load combination[13].

Level 1	→	Level 1
	→	

9.4.9 PGLD2

Purges all current loads.

Level 1	→	Level 1
	→	

Remarks:

NLD loads are not considered as loads, but as support definitions, and are not purged. This command works without confirmation. However, the command asks for confirmation when executed from the FEM2 (9.4.1) menu. The programmable version of the command is left-shifted available in the FEM2 menu.

9.4.10 PGRS2

Purges the calculation results of the current structure.

Level 1	→	Level 1
	→	

Remarks:

Can be used to save memory when files are saved, see SAVEFEM2 (9.4.13) command. This command works without confirmation. However, the command asks for confirmation when executed from the FEM2 (9.4.1) menu. The programmable version of the command is left-shifted available in the FEM2 menu.

9.4.11 NEWFEM2

Starts a new "empty" structure.

Level 1	→	Level 1
	→	

Remarks:

If the current structure is not "empty" you will be prompted to save the current structure before it is replaced by an "empty" one.

9.4.12 OPENFEM2

Opens a previously saved structure. You can select the file to open with a browser interface alike the HP48 choose box engine. The OPENFEM2 browser is much faster though.

Level 1	→	Level 1
	→	

Remarks:

If the current structure is not "empty" you will be prompted to save the current structure before it is replaced by the opened one. OPENFEM2 looks for files with the LCS extension (i.e. Truss.LCS) in the current directory.

9.4.13 SAVEFEM2

Saves the current structure as a file.

Level 1	→	Level 1
	→	

[13] in post processing mode

Remarks:

The LCS extension is automatically added by the program. The structure is saved in the current directory as a Library Data object. Illegal names (i.e. Truss Big) are allowed. Memory can be saved if you have the BZ compressor present on your HP48. Saved files will be BZ compressed first if the BZ option (with BZ¿ (4.3.72) command) is toggled on. An additional saved file size reduction can be accomplishes by purging the calculation results with the PGRS2 (9.4.10) command before saving the file. Note that then you will have to calculate the structure again after re-opening the file.

9.4.14 →FEM2

Stores a LCS file as the current structure.

Level 1	→	Level 1	
Library Data	→		overwrite current structure
0	→		erase current structure

Remarks:

Intended for programmable use. Overwrites or erases current structure. Checks for validity of the Library Data.

9.4.15 FEM2→

Recalls the current structure as a Library Data object.

Level 1	→	Level 1
	→	Library Data

Remarks:

Intended for programmable use.

9.4.16 →SNAM2

Stores a name string as the name of the current structure.

Level 1	→	Level 1
"string"	→	

Remarks:

Intended for programmable use.

9.4.17 SNAM2→

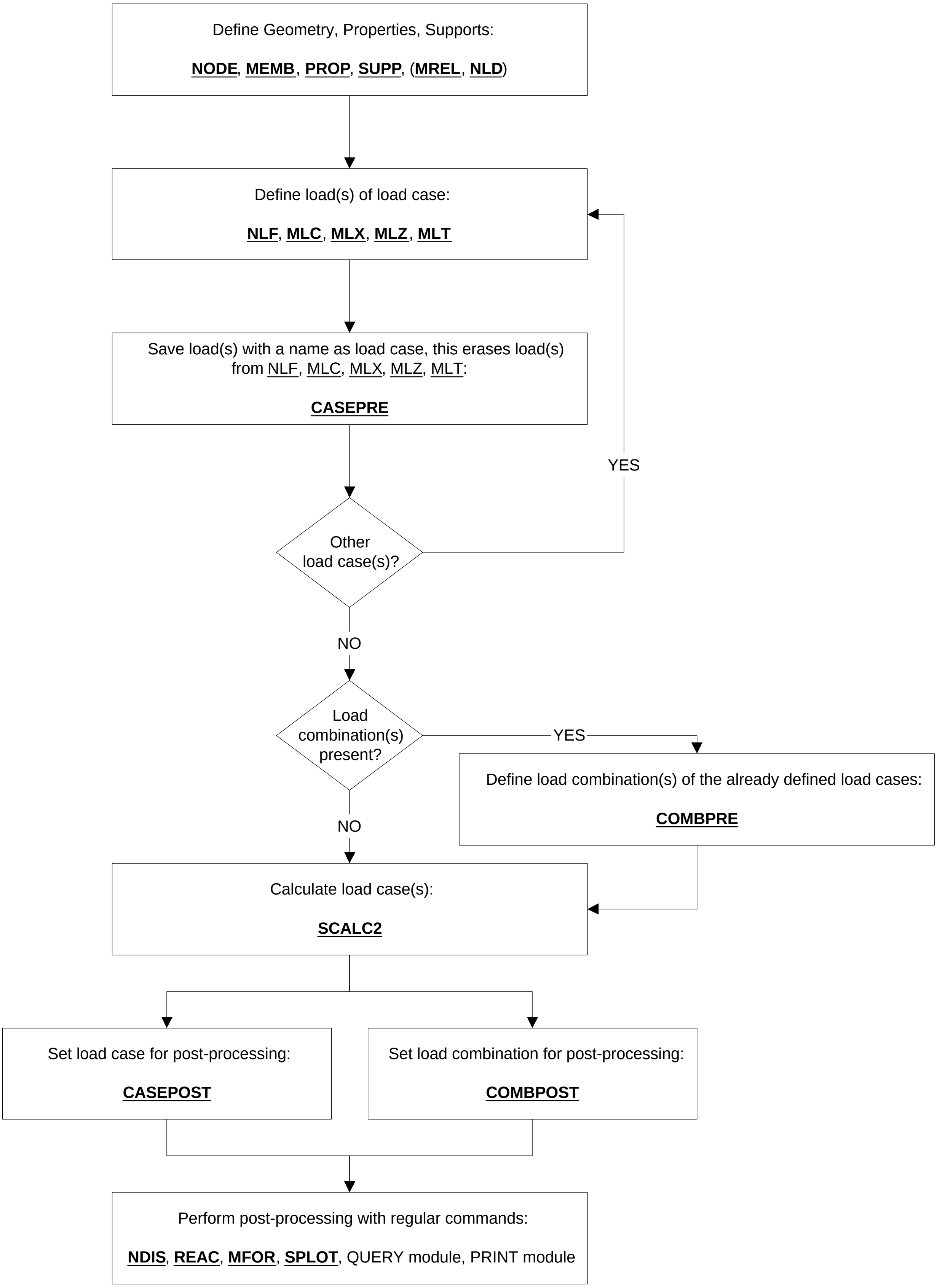
Recall the name string of the current structure.

Level 1	→	Level 1
	→	"string"

Remarks:

Intended for programmable use.

9.5 LCASE flowchart



10. TIPS AND TRICKS

10.1 Hand calculations

When performing calculations by hand you do not take extension into account. So, you are actually saying: members do not deform axially. When checking such calculations with FEM48 you have to be aware that you need to make the axial stiffness of the structure members very large[14] (make the cross sectional area very big), otherwise you will get answers that can not be compared with your hand calculation.

10.2 Inclined supports

FEM48 can not handle inclined supports by default. However, by using a trick you can! You replace the inclined support by a member that should be inclined in the same direction as the reaction force of the support. This member should be axially very stiff and have a release (4.3.25) at the node where the inclined support is wanted. At the other side of the member you define a pinned support. The axial force of the member is your reaction force! You may need to experiment a little bit before you get the axial stiffness and the length of the member ok.

10.3 Multiple structures

FEM48 does not check for multiple structures. So, you could take advantage of this and calculate two structures (unconnected by any member) at the same time.

10.4 Lack of fit

Lack of fit of a member can be modelled using an axial temperature load. For example, a member has a length of 5 meters and the lack of fit is 5 mm too short. Then the calculation of the axial temperature load is as follows:

$$L := 5\cdot m \qquad \Delta L := -5\cdot mm \qquad \alpha := 10^{-5}\cdot\frac{1}{K} \qquad \varepsilon := \alpha\cdot T_{axi} \qquad \varepsilon := \frac{\Delta L}{L} \quad T_{axi} := \frac{\Delta L}{\alpha\cdot L} \quad T_{axi} = -100K$$

10.5 Random temperature loads

As opposed to the example provided at the MLT (4.3.43) command description, temperature loads are often discontinuous and/or non-linear. Even so, such temperature loads can also be splitted into a normal strain/stress and curvature/bending stress part. After splitting you will also get a third component[15], which will result in neither elongation/shrinkage nor bending but which will result in stresses[16]! The formulas used for splitting any temperature load are given below, where x is zero at the top of the section and is positive downwards and $\Delta T(x)$ is your temperature function:

$$\Delta T_{axial} = \frac{1}{Area}\cdot\int_0^{height} \Delta T(x)\cdot width(x)\,dx \qquad \Delta T_{difference} = \frac{height}{Inertia}\cdot\int_0^{height} \Delta T(x)\cdot width(x)\cdot\left(x - z_{top}\right)dx$$

$$\Delta T_{bending}(x) = \frac{\Delta T_{difference}}{height}\cdot\left(x - z_{top}\right) \qquad \Delta T_e(x) = \Delta T(x) - \left(\Delta T_{axial} + \Delta T_{bending}(x)\right)$$

10.6 Parallel list processing

This works with: NX, VX, MX, UXX, RYX, UZX, NTAB, VTAB, MTAB, UXTAB, RYTAB and UZTAB.

---

[14] Only if there are members in the structure that are loaded axially.
[15] Here named $\Delta T_e(x)$.
[16] Deformation due to the “third component” will only occur at the free end(s) of a section.

10.7 Programming

Since FEM48 is completely programmable, you can easily customize it to your needs. Below a UserRPL program is shown that calculates a stress table for the desired member[17]. Calculated are shear stress and the combined axial and bending stress at top and bottom of member. The QUERY module needs to be present.

Stack syntax:

Level 1	→	Level 1
integer	→	[[matrix]] $_{n\times 4}$

Remarks:

The UserRPL program needs a data matrix with the name PROP3 as a user variable to be present in the current (or in a parent) directory. This matrix contains additional section properties, where each row of PROP3 belongs to the same row of PROP (4.3.16, FEM48 command). Thus the PROP3 matrix should have a number of rows equal to the PROP definition used by FEM48.

Column one of PROP3 should contain the area used to calculate shear stress. Note that this can be different from the area of the PROP definition (with steel H beams for instance)! If this is the same though, you could just enter zero (0) and the program will use the area of the PROP definition (see code below). Column two of PROP3 should contain the distance from the elastic line to the top of the section, z-top. Column three of PROP3 should contain the distance from the elastic line to the bottom of the section, z-bottom.

The calculation results are: [[x τ σtop σbot]]

Note that the program was coded for educational purposes, not for speed or small size!

```
«
      AV? 0    0    0     0     0
  →
  memb av prop area mrtop mrbot table              @LOCAL VARIABLES
  «
   0 AV¿                                           @NO AUTOVIEW
   "Stress Table:

[[ x τ σtop σbot ]]

"
   3 DISP                                          @DISPLAY CALC STATUS
   MEMB→ memb 3 * GET                              @PROP#
   'prop' STO
   PROP3 prop 3 * 2 – GET                          @AREA
   IF
    DUP 0 ≤                                        @HANDY TO TYPE LESS INPUT
   THEN                                            @USE STD AREA
    DROP
    PROP→ prop 3 * 2 – GET                         @AREA OF FEM48 PROP
   END
   'area' STO
   PROP→ prop 3 * 1 – GET                          @IY
   DUP
   PROP3 prop 3 * 1 – GET                          @GET ZTOP FROM PROP3 VARIABLE
   / 'mrtop' STO
   PROP3 prop 3 * GET                              @GET ZBOT FROM PROP3 VARIABLE
   / 'mrbot' STO

   memb VTAB                                       @CALULATE VTABLE
```

[17] Note that with the introduction of version 5.2, stress calculations are included in the QUERY module. The example is kept for educational purposes.

```
  memb NTAB                                      @CALULATE NTABLE
  memb MTAB                                      @CALULATE MTABLE

  DUP →COL DROP                                  @INITIALIZE TABLE
  DUP DUP 4 COL→
  'table' STO
  1 KEYP→                                        @ROWS OF TABLE EQUALS KEYPNTS
  FOR j
   OVER                                          @COPY OF NTABLE
   j 2 * GET                                     @GET N
   area /                                        @AXIAL STRESS
   OVER                                          @COPY OF MTABLE
   j 2 * GET                                     @GET M
   mrtop /                                       @BENDING STRESS
   -                                             @COMBINE TO σTOP
   table
   j 4 * 1 - ROT PUT                             @PUT IN TABLE
   'table' STO

   OVER                                          @COPY OF NTABLE
   j 2 * GET                                     @GET N
   area /                                        @AXIAL STRESS
   OVER                                          @COPY OF MTABLE
   j 2 * GET                                     @GET M
   mrbot /                                       @BENDING STRESS
   +                                             @COMBINE TO σBOT
   table j 4 * ROT PUT                           @PUT IN TABLE
   'table' STO

   3 PICK                                        @COPY OF VTABLE
   j 2 * GET                                     @GET V
   area /                                        @SHEAR STRESS
   table j 4 * 2 - ROT PUT                       @PUT IN TABLE
   'table' STO
  NEXT
  DROP2 DROP                                     @DROP NTABLE VTABLE MTABLE
  table                                          @RECALL STRESS TABLE
  IF
   RND?                                          @USE ROUNDING?
  THEN
   RVAL→  RND                                    @GET ROUNDING VALUE AND RND
  END
  IF
   TAG?                                          @USE TAGS?
  THEN
   SNAM→ " " + memb +                            @TAG WITH NAME, MEMBER#,
   " x τ σtop σbot" +                            @AND ARRAY FORMAT
   →TAG
  END
  av AV¿                                         @RESTORE AV TOGGLE
 »
»
```

11. MISCELLANEOUS

11.1 Memory usage

A good guesstimate, with memory equaling the amount of free memory needed in bytes and nodes equaling the number of nodes in the structure, is:

$$\text{memory} = 1.1 \cdot \left[15 + 8 \cdot (3 \cdot \text{nodes})^2 \right]$$

which about equals:

$$\text{memory} = 79 \cdot \text{nodes}^2$$

11.2 Speed

FEM48 speed depends mostly upon the library version (standard or compressed) and the port it is stored (uncovered or covered). A good rule of thumb is:

- compressed versions execute 20% slower than the standard versions
- when stored in a covered port (2 or greater) the execution is about 13% slower than when stored in an uncovered port (0 or 1)

Measurements were done for the SCALC routine.

11.3 Limitations

Although FEM48 is very capable it does have its limitations:

- FEM48 should only be used for the calculation of structures which have relatively small displacements.
- Deformation due to shear force is not taken into account. For rectangular sections, the influence of this deformation will become noticeable when members are shorter than 5 times their height.

12. AUTHOR AND CREDITS

12.1 Author

FEM48 has been created by Caspar Lugtmeier.
Contact the author[18]: clugtmeier@gmail.com

12.2 Credits

I would like to thank the following people:

- All users who gave support, asked questions and provided bug reports (and thus motivated me to improve FEM48)
- Gjermund Skailand for allowing the use of his Cholesky matrix solver
- Hector Bernardo for translating this Reference Manual to Spanish
- Oscar Fuentes Fuentes for creating a Spanish Examples Manual
- Edwin Córdoba for porting FEM48 v4.4 and previous to the HP49 (and doing this more than once!)
- Alex Tomanovich and Alexander Reif for beta-testing and giving suggestions for improving FEM48
- Mika Heiskanen and Jan Brittenson for creating Jazz
- Sebastien Carlier and Christoph Gießelink for creating and improving EMU48
- Eric Rechlin for adding FEM48 to his wonderful site www.hpcalc.org

[18] In case of questions: after reading the manual…

13. INSTALLATION

13.1 Installation

The FEM48 library modules can be installed in any available port on the calculator. However, they work fastest from a non-covered port, that is port 0 or 1. To install the FEM48 library modules on your calculator do the following:

- Transfer the library module to your calculator and place the library module on the stack.
- Place the port number you wish to store the library module in on the stack and press the STO button.
- Warmstart the calculator by pressing ON and C simultaneously.
- Purge the variable which still contains the library module.

When in doubt, please refer to chapter 28 of the HP48 User's Guide.

13.2 Characteristics

Module	Library ID	HP48 (HP49/49+/50 not mentioned)				optional	misc.
		"standard"		"compressed"			
		bytes	checksum	bytes	checksum		
FEM4x	1605	29367	19612d	22586.5	13375d	no	-
QUERY	1606	15662	607d	12896	40780d	yes	nullnamed
WIZRD	1607	10692.5	37226d	8992	52689d	yes	nullnamed
PRINT	1608	4689.5	56977d	3082.5	56337d	yes	nullnamed
MOVLD	1604	2386.5	13180d	1987.5	57493d	yes	nullnamed
LCASE	1603	10460.5	21279d	9105	41933d	yes	nullnamed
	Total	73276	Total	58663			

Flags used:

-20, -21 and –22 (SCALC, SCALC2 and MOVLD, all cleared)

Variables used:

- FEM_in, FEM_ck, FEM_out and FEM_cfg (FEM48 module, in hidden directory)
- LCASE_in, LCASE_ck, LCASE_out, LCASE_comb, LCASE_geo and LCASE_name (LCASE module, in hidden directory)
- FEM_prop2 (QUERY module, in hidden directory)
- ↑NDIS, ↓NDIS, ↑REAC, ↓REAC, ↑MFOR and ↓MFOR (MOVLD module, current directory)

Language used:

- SystemRPL (99%) and Assembly (1%)

13.3 Statistics

Module	number of XLIBS		
	named[19]	nullnamed[20]	total
FEM48	84	125	209
QUERY	48	54	102
WIZRD	13	26	39
PRINT	11	9	20
MOVLD	2	6	8
LCASE	17	34	51
Total	175	254	429

[19] commands
[20] subroutines

14. VERSION HISTORY

Version 5.3 SR7

Included F.USR: UserRPL program that plots N, V, M and uXZ for the whole structure
FEM49+: Updated FEM49+ / FEM50 main module to enable access of F.USR by right-shift SCALC (note that you can replace the included F.USR by your own version)

Version 5.3 SR6

FEM49+: bugfix of SPLOT routine

Version 5.3 SR5

FEM49+: adapted to use bigger screen height of HP49g+ and HP50g
FEM49+: enabled FAST mode for HP49g+ and HP50g (overclocking CPU to 203 MHz)

Version 5.3 SR4

bugfix: Huber Hencky stresses (top and bottom) were calculated incorrectly (4*τX^2 should be 3*τX^2)
change: New load case text modified to: Save As New Case

Version 5.3 SR3

bugfix: HP49 MOVLD module has been modified to accept ZINT's as arguments
bugfix: HP49+ library MOVLD access from FEM main module has been fixed

Version 5.3 SR2

bugfix: in previous version CASEPOST and COMBPOST deleted PROP2 definitions, now fixed (only LCASE module has changed)

Version 5.3 SR1

bugfix: fixed small font and keywait bug in HP49 WIZRD module (for display of Rectangular (multiple) wizard

Version 5.3

Also available as FEM49, ported by the author of FEM48

FEM48 module

- changed default configuration settings
- changed menulabel order of INPUT menu when LCASE mode is active

Version 5.2

FEM48 module

- improved reaction to bad input with input prompts
- ON key now cancels input with inputline prompts, previous it cleared the command line
- fixed a small bug in MATV command
- file format has changed (due to PROP2 definition of QUERY module)

QUERY module

- added stress calculations (shear, combined axial+bending, Huber-Hencky)
- modified QUERY menu (note shifted plane menukey actions!)

PRINT module

- fixed a small bug in STRV command

Version 5.1

FEM48 module

- changed default configuration settings

MOVLD module

- moving loads are now possible on all kind of structures (was only on beams)

WIZRD module

- added MLCG, MLXG and MLZG commands: generate MLC, MLX and MLZ loads for the global system
- updated menugrobs

Version 5.0

LCASE module

- new load case and combinations module
- now you can save loads as load cases and make combinations of load cases
- includes updates of FEM48 commands like SAVEFEM2, PGLD2, SCALC2 etc

FEM48 module

- added temperature loads option with command MLT
- added display of choice position and choice list length to FBROW browser
- SCALC now always uses Cholesky solver
- added recall state of toggle commands (e.g. now BZ¿ sets state and BZ? recalls state)
- CHOL command removed
- LCAS command added
- modified SINFO display
- added protection for FBROW for large header strings
- removed autocalculation with SPLOT command (when DFOR¿ is toggled on)

QUERY module

- modified NPLT, VPLT, MPLT, UXPLT, RYPLT and UZPLT commands, which makes it possible to plot N, V, M, etc lines for more than one member in one plot (continuous beams!)
- now also scrolling along plotted graphs, with displayed values under plot (trace mode)
- removed batchplot facilities
- removed QMEM commands, now member provided by user for each command (faster)
- improved scaling of constant value plots (now in middle of screen)
- improved reaction to ON (cancel) key
- added optional tagging of stack results

WIZRD module

- added Ix to SED48 link (was only choice of Iy and Iz), order is now: Ix Iy Iz, with default position on Iy
- bugfix: now Iz selection in SED48 link chooses Iz correctly (was Iy)

PRINT module

- all non-integers are now displayed using stack format (e.g. 3 ENG)

MOVLD module

- MOVLD command is now available from the FEM menu (Left-Shift SCALC or SCALC2)
- now a nullnamed library

Version 4.4

Also available as FEM49, thanks to Edwin Córdoba

FEM48 module

- fixed bug in OPENFEM command
- textual change for the NODES inpuline command

Version 4.3

Also available as FEM49, thanks to Edwin Córdoba

FEM48 module

- added Cholesky matrix solver, thanks to Gjermund Skailand, this is written in assembly and is much faster and more memory efficient than the HP solver: use this as default solver!
- removed Gauss matrix solver (obsolete due to Cholesky solver)
- added grobs for spring supports
- improved nodal displacement loads handling:
  NLD loads now overrule supports: no more need to unrestrain a degree of freedom before giving it a NLD
  improved REAC output: even with NLD loads, each node with a reaction is now only mentioned once
- added UserRPL interface for FEM browser: FBROW

WIZRD module

- made the structure wizards more uniform, now all wizards use length per span/bay input
- added multiple (!) rectangular section wizard, for rectangles, T sections, I sections, etc

Version 4.2

Also available as FEM49, thanks to Edwin Córdoba

MOVLD generator

- newly available Moving Loads Generator
- define load system of an unlimited number of concentrated member loads and the interval + stepsize
- outputs the NDIS REAC MFOR minimums and maximums in the current directory
- when defined the user program MOVLD.USR is run after each step, so you can add your own postprocessing!
- available as separate library: MOVLD with number 1604

FEM48 module

- added optional node & member numbering in SPLOT
- major change in internal file-management: more speed (thus file-format has changed!)
- added cache for input check: more speed
- improved code for input check routines
- added custom browser for OPENFEM: more speed
- added optional GAUSS solve, now you can solve bigger problems (though it is very slow compared to "standard" routine)
- added programmable STYPE command: recalls structure type string
- added confirmation for PGLD and PGRS in menus only
- removed inverse status display
- removed "shift" indicators when calculating
- moved STRV commands to the PRINT module
- rewritten YesNo routine, now reacts to YES/NO menulabels and ENTER/ON keys
- bugfix: now NLD input is not overwritten after calculation
- bugfix: now ticking clock is properly suspended for ERASEFEM command
- bugfix: now 0 ->FEM works ok
- XLIBS have changed from v4.1 -> please check your own UserRPL progs with FEM commands
- some minor changes

WIZRD module

- added Properties generator: Rectangle, Tee, Circle, Circular Ring, Numerical, SED48 Database (!)
- About SED48 Database link: Get the latest SED48 **version 1.2** database library at my website, WIZRD looks for properties { "A" "Iy" "E" } or { "A" "Iz" "E" }, routine for conversion mm N <-> m kN is built-in
- added Selfweight generator (!)
- added Pre/Post stressing cable generators
- now completely SysRPL like rest of modules

QUERY module

- keypoints of 2=<keyp=<131 now allowed (was 3=<keyp=<50)

General

- removed all unsupported entries, this to improve portability of code

Version 4.1 pre-release

- made *all* commands programmable(!), so lib 1604 (see v4.0/3.3) will not be needed
- note: *all* commands (ready for programmable use) available from FEM menu (just try LS or RS toggle labels to see what I mean by *all* commands :)
- added some extra commands for easy programmable file management ->FEM, FEM-> store and recall current file ->SNAM, SNAM-> store and recall file name note: 0 ->FEM erases the current FEMpar file ! (in the hidden directory)
- added configurable external matrixwriter and text editor option on second row of FEM menu (used by AV)
- added delay option between batchplots (wait for key or pause for X seconds between plots)
- STYPE now only recalls current type, SFRAM and STRUS set type
- the CKFEM program is now part of the FEM48 library
- small code improvements
- removed some unsupported plot entries, speeds up plotting
- all plots now made without removing the menu, so no more keypress to resume calculator use (this also eases programmable use)
- removed ->PICT option, added ->STK option for plots
- renamed TSIGN option to [upwards arrow]M and it now only works on the moment plot
- bugfix: nodal coordinates displayed with PRINT module now in X and Z (was X and Y)
- bugfix: MINFO (QUERY menu) text labels now OK (MLU to MLX, MLT to MLZ)
- bugfix: when QUERY plots are constant, both the + and - labels now have the same value
- removed recalculation warning
- removed confirmation of PGLD and PGRS commands
- size: you might want to buy an extra RAM card :(
- but the libs are also available in a compressed version :)

Version 4.0 pre-release

- added displacement loads for nodes (NLD) format: [[Node UX UZ RY]], notes: UX/UZ/RY =0 : no displacement load (but also no support!), be carefull with mix and match with the SUPP command!, a degree of freedom (i.e. UZ of node 10) can NOT have a SUPP and NLD at the same time!
- added possibility of transverse loads on members with end releases
- added member load local X trapezoid MLX
- removed MLUNI
- renamed MLT to MLZ
- renamed NL to NLF
- restructured INPUT and RSULT menus
- programming lib (1604) not available yet
- lib and modules also "BZ compressed" available, this shaves +/- 25% off of the size and is about 10% slower, the uncompressed and compressed modules can be mixed and matched, the presence of BZ is not needed, the UBZ decompressor is built-in
- made program CKFEM available: checks version/availability of loaded FEM48 modules
- no manual yet, please use manual of version 3.2 and read this changes.txt file carefully
- Note: Not all features that were "advertised" have made it to the final release. It would also still be possible to create a significant speed increase by completely rewriting FEM48. However, the resulting *final* package of FEM48 is probably the most powerful and easy to use Structural Analysis suite available to date for any (HP) calculator.

Version 3.3 pre-release

- modular(!) design:

- lib 1605: FEM library (main library with interface)
- lib 1606: Member Query menu (nullnamed lib, QUERY in RSULT menu)
- lib 1607: Structure Wizard menu (nullnamed lib, LeftShift-INPUT)
- lib 1608: String output / Print menu (nullnamed lib, LeftShift-RSULT) * NEW FEATURE *
- lib 1604: Command lib (for programming FEM, not available yet)
- All menus accessible through FEM menu from main FEM lib (except lib 1604)
- FEMpar variable now stored in hidden directory
- RNDing when recalling results, thus now member query with full precision
- renamed commands, most notably: CONST > SUPP, SUPPR > REAC, NOCO > DET
- Changed input for MREL: 1=release, 0=fixed to node
- Changed input for SUPP: 0=free, 1=fixed, <0=spring
- Added error message for UZ and RY member query when member has release at first node (BTW: still no transverse loads on members with MREL possible!)
- Added PGLD (purge loads) command
- Added REVIEW (RightShift-DownArrow) for all key planes to see all unshifted, right-shifted and left-shifted softkey actions (USE THIS!)
- Added numeric labels for queried member and no. of keypoints
- Replaced ML lineon routine by standard ML lineon command
- Changed support grobs
- Seperated forces and displacement output/plots in query menu (MFORC,MDISP > N,V,M,UX,UZ,RY)
- No more "Undefined Loads" error when plotting just the geometry
- Restyled all INFO commands
- Restyled various menus

Version 3.2

BZ compressed

- see 3.2pro
- compressed subroutines so lib is much smaller (-7 kB !)
- not programmable!
- adds support for the FWZRD library (LS INPUT in FEM menu)

pro

- see 3.2std
- now also available as FEM49, thanks to Edwin Córdoba who ported the FEM48 library to the HP49

std

- added support for the EDITB command of Meta Kernel
- added fast DOVARS routine for OPEN command
- fixed a small bug with respect to a call to FEM_cfg (not a calculation bug)
- now almost all commands are programmable (Programmable Edition no longer separately available)
- Student Edition no longer available
- updated manual to reflect the changes

Version 3.1

pro

- see 3.1std
- fixed bug in batchplot routines for the NPLT, VPLT, MPLT, UXPLT, UZPLT and RYPLT commands
- now the member number is displayed on the QUERY plots
- modified MINFO display

std

- added recalculation warning for CALC
- renamed ERASEFEMcfg to ERASEFEM
- changed some Yes/No routines and displays (SAVE overwrite)
- changed STYPE display
- improved the manual

Version 3.0

pro

- see 3.0std
- added X-coordinate to tagged output of XFORC and XDISP
- added batchplots for the NPLT, VPLT, MPLT, UXPLT, UZPLT and RYPLT commands
- now the QUERY menu is accessible from the RSULT menu
- restyled QUERY menu, now MFORC and MDISP available as shifted commands from XFORC and XDISP menukeys
- restyled MINFO, the "look" of the display now matches the rest of the library
- number of keypoints used is now displayed by MINFO

std

- FILE FORMAT HAS CHANGED: .FEM files of previous versions can not be used with version 3.0, use the in the FEM48 package, included conversion program, see "File conversion.txt" for more details
- added member end release option for bending moment
- added "busy indicators" to indicate activity when FAST is used
- restyled SINFO and input check display output, the "look" of the display now matches the rest of the library
- renamed TYPE to STYPE
- improved the manual, now a separate Reference Manual and Examples document
- bug fix: now autocalculation during DFOR plots uses FAST (when toggled on) too

Version 2.1

pro

- see 2.1std
- renamed commands: uXPLT > UXPLT, uZPLT > UZPLT, thetaYPLT > RYPLT

std

- added FAST option, now you can reduce the execution time of calculations by approx 10%
- renamed commands: MLTRAP > MLTRP, MEFORC > MEFOR, MEDISP > MEDIS, INFO > SINFO

Version 2.0

pro

- see 2.1std
- first release of the professional version of FEM48: includes beam analysis
- file format identical to version 2.0std (NOT compatible with file format of older versions)
- improved and modified the docs as to apply to both versions
- as of now the version numbers of the std and pro version will be identical to make an easy comparison possible

std

- FILE FORMAT HAS CHANGED: .FEM files of version 1.2 can not be used with version 2.0, use the in the FEM48 package included conversion program, see "File conversion.txt" for details
- added spring constraints, now you can calculate structures with displacement and rotation spring supports, so the format of CONST has changed!
- now the filename is saved in the FEMpar file, this avoids having to enter the filename again when saving your work

- now NDISP also outputs the node number, this to enable easy viewing outside the matrixwriter
- added support for the BZ compressor, available in the FILE menu, this greatly reduces the saved file-size
- before calculation starts old results are deleted from FEMpar to save memory which may be needed during calculation
- now trusses and frames are plotted with different joint grobs for easy recognition of structure type
- added calculated indicator (C) to INFO display
- added second INFO menulabel to INPUT menu
- now the open file choose box is full screen
- added confirmation to PGRES command
- moved PGRES menulabel from 2nd to 1st menurow of FILE menu
- added TYPE rcl (right-shift TYPE)
- renamed MLLIN to MLTRAP (from linear to trapezoid)
- renamed MFORC to MEFORC (Member End FORCe)
- renamed MDISP to MEDISP (Member End DISPlacement)

Version 1.3std and 1.3pro

- released only as beta-version

Version 1.2std

- added file browser, now you can save (and open) your structures with a unique name
- improved/modified the docs

Version 1.1std

- changed INPUT menu, now displayed faster
- removed CFGR menu, now all config commands available in neccessary menus
- now multiple occurences of a node in the nodal loads array is supported, as was already the case with member loads
- added status display for input check
- added autoview recalled input/output with built-in matrixwriter or MATRIX
- added nodes and constraints option for plotter
- small code improvements
- modified the docs

Version 1.0std

- first full release
- changed interface type from browser-based to menu-based (faster and more "open" interface)
- added plotter
- added linear (trapezoid) member loads
- changed all loads array format
- removed the file open/save option, now works with variable: FEMpar
- removed concentrated member moment load bug
- removed releases option, didn't work out (truss option still available)

Version 0.2beta

- released february '99
- initial release to the public