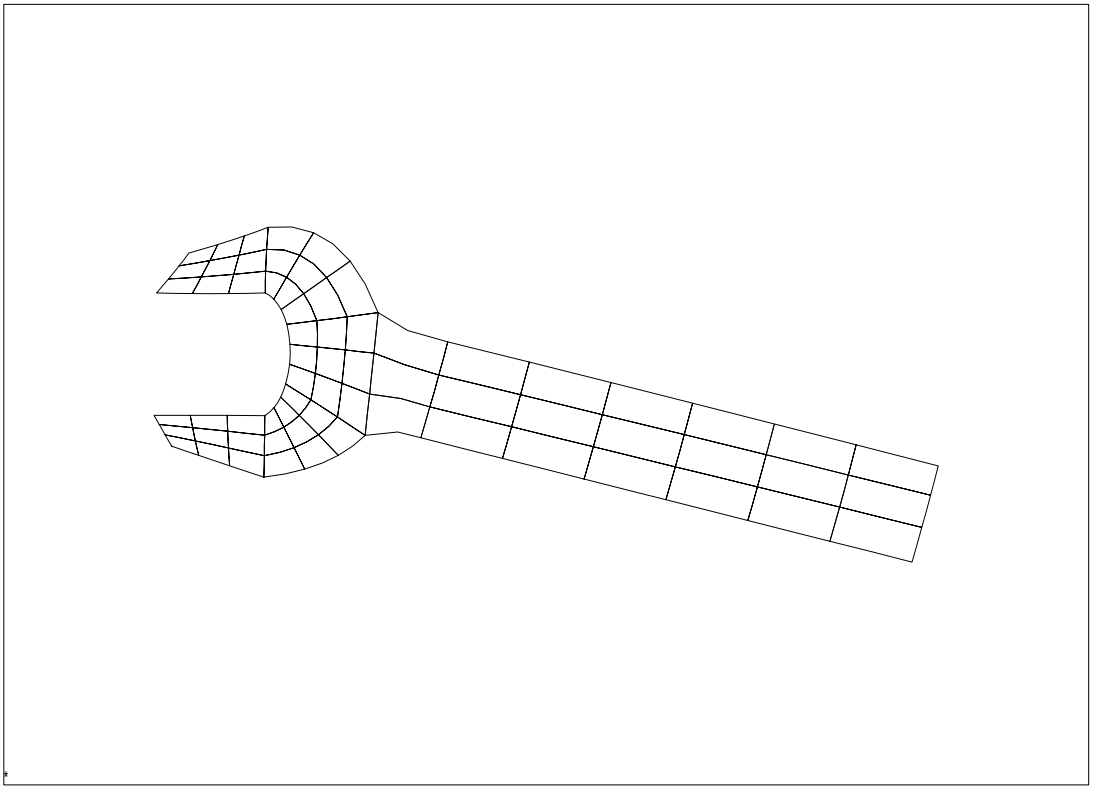
Z88

The compact Finite Element System



Version 11.0

Z88

A modular, compact und fast
Finite Element Program in ANSI-C
for all Windows and UNIX computers

Both the UNIX and the Windows versions
are Freeware and are protected by the
GNU General Public License

Composed and edited by
Professor Dr. Frank Rieg
Lehrstuhl Konstruktionslehre und CAD
(Chair for Engineering Design and CAD)
University of Bayreuth, Germany

© Version 11.0 May 2004

WELCOMES TO Z88 !

Z88 for UNIX and Windows is Freeware and is protected by the GNU General Public License, see next pages.

Z88 is compact and fast and was developed for PCs initially. Today, Z88 runs properly on LINUX and Windows (starting with Windows95) PCs, powerfull UNIX workstations and large vector computers, however. Simple to compile and to install. Handling is simple. Z88 comes with context sensitive online-help and a user- friendly command processor. Successfully proved by some hundreds of commercial installations and countless UNIX- and LINUX installations. For static calculations in mechanical engineering and building & construction industries. Absolutely transparent for the user as input and output are handled by text files.

What makes the new version 11.0 even better than the approved version 10.0 ?

- With the new OpenGL plot program Z88O the Z88 system enters new dimensions. You may illuminate a structure with three different light sources or plot with hidden lines, both the undeflected and the deflected structure. You may plot stresses and X, Y and Z deflections with a color range - like the expensive professional FEA programs. You may plot a limited range of nodal or element numbers - a nice feature especially for large structures. Of course, you may define your choice of colors, the light features and the fonts (for UNIX) in the parameter files Z88O.OGL (for Windows) and Z88.FCD (for UNIX). This is the main new feature in versions 11 but it took very tricky changes in the computations of stresses along with a lot of studies and experiments. The well-known plot program Z88P stays included in the Z88 system - so you have the choice between two plot programs.
- The COSMOS converter can now convert NASTRAN files in addition, however. This is important for the new Pro/ENGINEER Wildfire which does not support COSMOS files any longer.

Z88 has deliberately been designed as a compact and fast system. Thus, its use is restricted to static calculations. Z88 does not want to compete with professional FEA programs for workstation or mainframes which can do really everything, but are hardly payable and complicated to operate. While you are still puzzling about installation and start of some programs of this genre also in the PC class, you have already calculated the first examples with Z88. And the online-help is always only one keystroke or mouse click away. The Z88 system may operate with English or German language depending on your setting (ENGLISH or GERMAN) in the file Z88.DYN.

If you already have FEA experiences, you can start at once. If you are a beginner in this area, i would recommend secondary literature. Here are a few choices:

- *Zienkiewicz, O.C.; Taylor, R.L.: The Finite Element Method, Volumes 1-3, 5th edition, Butterworth- Heinemann and John Wiley & Sons, 2000*
- *Bathe,K.J.: Finite Elemente Procedures. Prentice Hall, 1995*
- *Schwarz, H.R.: Methode der finiten Elemente. Teubner publishing house, 3rd edition, Stuttgart 1991 (in German language)*
- *Rieg,F.; Hackenschmidt, R.: Finite Elemente Analyse für Ingenieure. Hanser- Verlag, München Wien 2003, 2nd edition (in German language)*

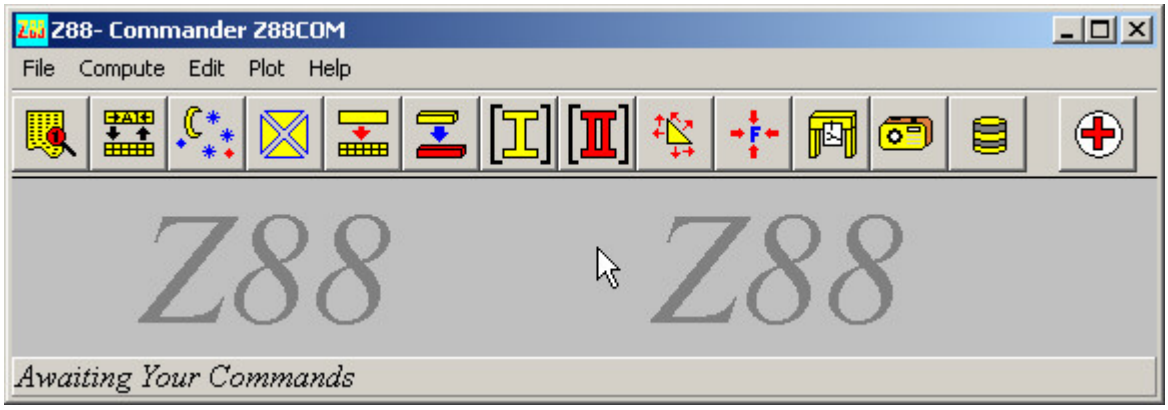
If you like programming, especially such technical programs as FEA or CAD programs you may want to consult the following book:
Rieg,F.; Hackenschmidt, R.: Softwaretechnik für Ingenieure. Hanser- Verlag, München Wien 2001 (in German language)

If you improve Z88 please give me your feedback. If you want to compile Z88 for UNIX any C compiler along with a *Motif* library should work fine - i've tested the GNU *gcc* and the C compilers from SGI and HP. If you want to compile Z88 for Windows any kind of C or C++ compiler should work fine - i've tested the free *LCC* and the C/C++ compilers from Microsoft, Borland and Watcom.

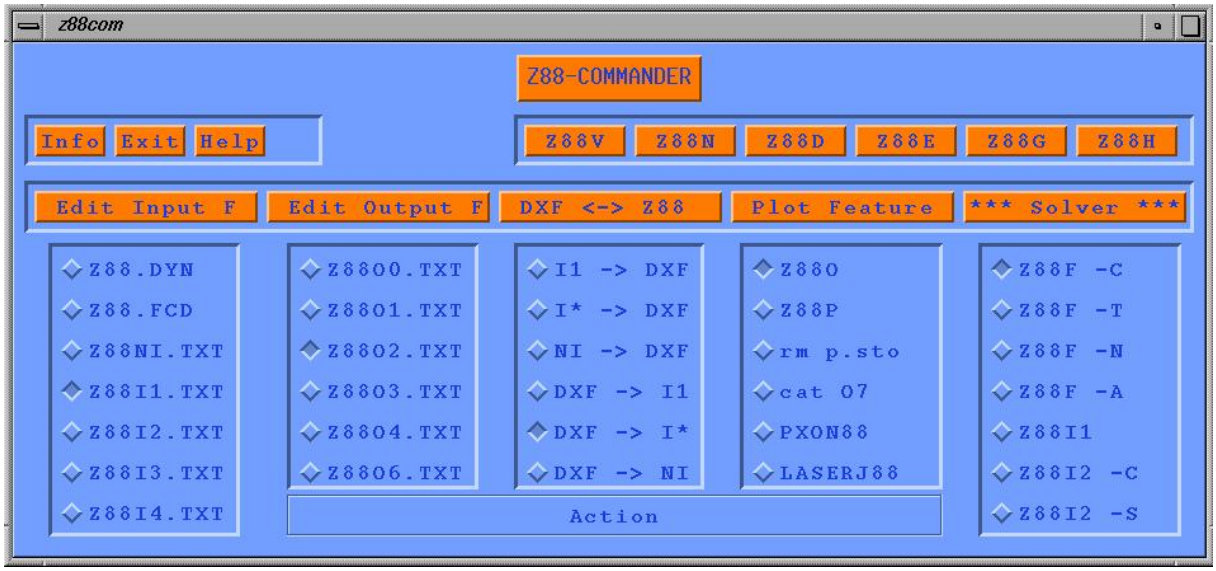
And because Z88 is bound to the GNU General Public License you are to present your improvements and modifications to the public including the source code - that's a point of honor. Promote the idea of the free software according to GNU-GPL !

Professor Dr. Frank Rieg
Lehrstuhl Konstruktionslehre und CAD
(Chair for Engineering Design and CAD)
University of Bayreuth, Germany
frank.rieg@uni-bayreuth.de
www.uni-bayreuth.de/departments/konstruktionslehre

Bayreuth, May 2004



The Z88 Commander for Windows



The Z88 Commander for UNIX and LINUX

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and

disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will

be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

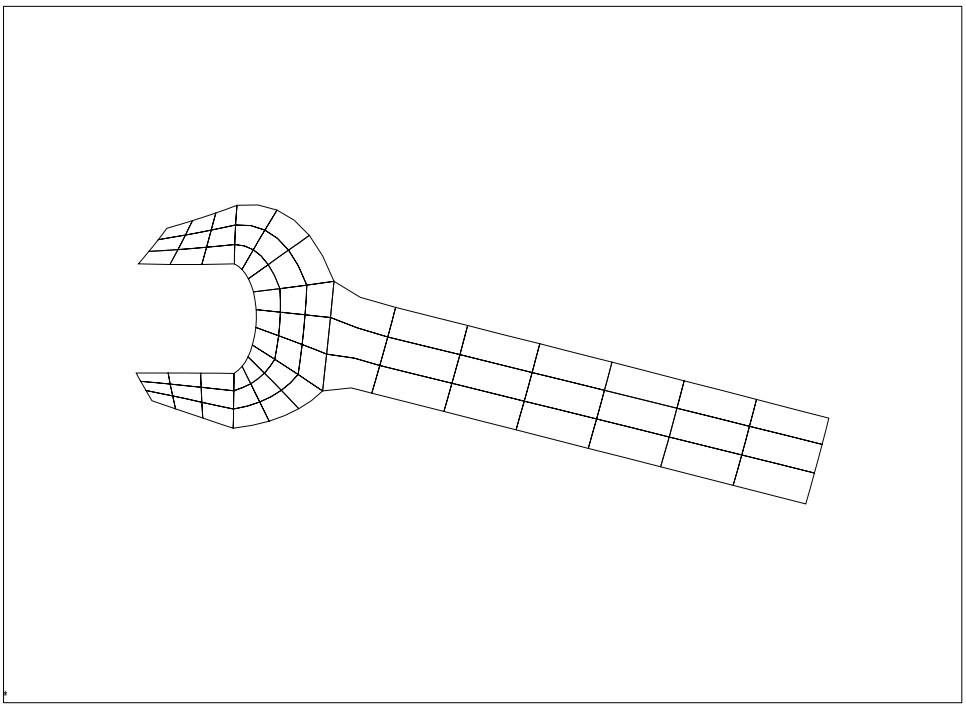
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

TABLE OF CONTENTS

1 THE FINITE ELEMENT PROGRAM Z88



1.1 GENERAL OVERVIEW FEA PROGRAM Z88

The Z88 philosophy:

+ Fast and compact: Developed for PC, no ported mainframe system
+ Flexible and transparent: Controlled by text files
+ "Small is beautifull" - a modular system vs. monolithic monsters
+ native UNIX and Windows programs, no emulation
+ UNIX and Windows programs use the same computing kernels
+ Full data exchange from and to CAD systems with DXF-Interface
+ mesh import from Pro/ENGINEER
+ Context sensitive online-help under Windows and UNIX
+ No copy protection, no annoying passwords
+ Simplest installation: No subdirectories, no change of system files
+ Under UNIX: Automatic control and cumulative runs possible

Notes:

Always compare FE calculations with analytical rough calculations, results of experiments, plausibility considerations and other tests without exception!

Keep in mind that sign definitions of Z88 (and also other FEA programs) differ from the usual definitions of the analytical technical mechanics from time to time .

Z88 is a complex computer program. How Z88 deals with other programs and utilities etc. is not predictable. I cannot give any advice and support here! You should switch off at first all other programs and utilities. Run Z88 "purely" and then start further programs step-by-step. Z88 uses only documented operating system calls of Windows and UNIX !

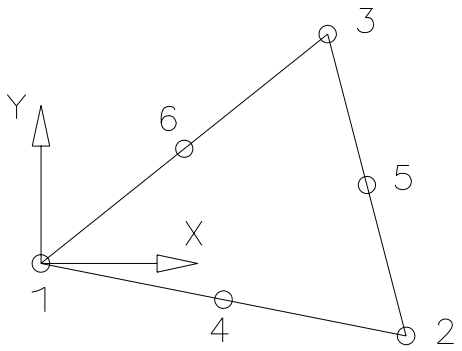
Summary of the Z88 element library:

(You will find the exact description of the element library in chapter 4.)

Twodimensinal problems: Plane stress, plates, beams, trusses

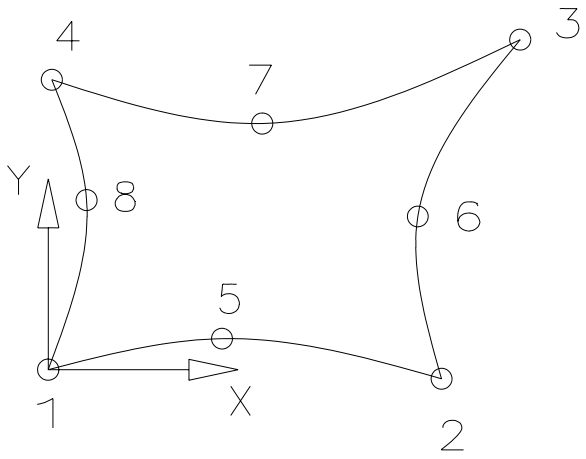
Plane Stress Triangle Element No. 3

- Shape functions quadratic
- Quality of displacements very good
- Quality of stresses in the center of gravity good
- Computing effort: average
- Size of element stiffness matrix: 12 * 12



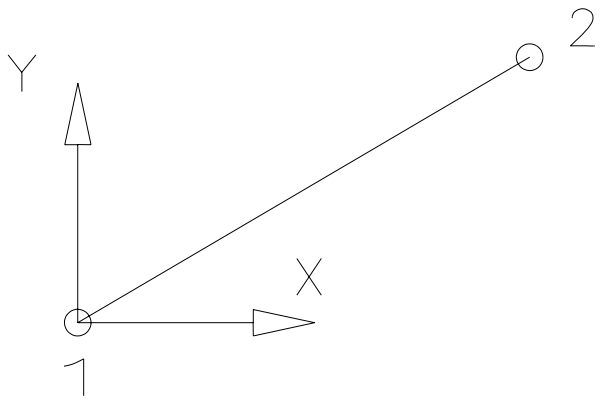
Plane Stress Isoparametric Element No. 7

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Quality of stresses in the Gauss- points very good
- Quality of stresses in the corner nodes good
- Computing effort: High
- Size of element stiffness matrix: 16 * 16



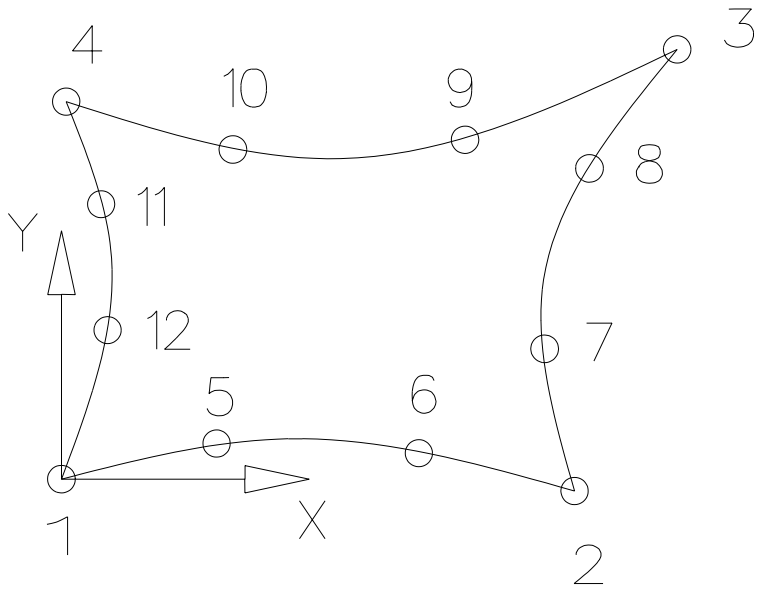
Truss No. 9

- Linear function
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Minimal
- Size of element stiffness matrix: 4 * 4



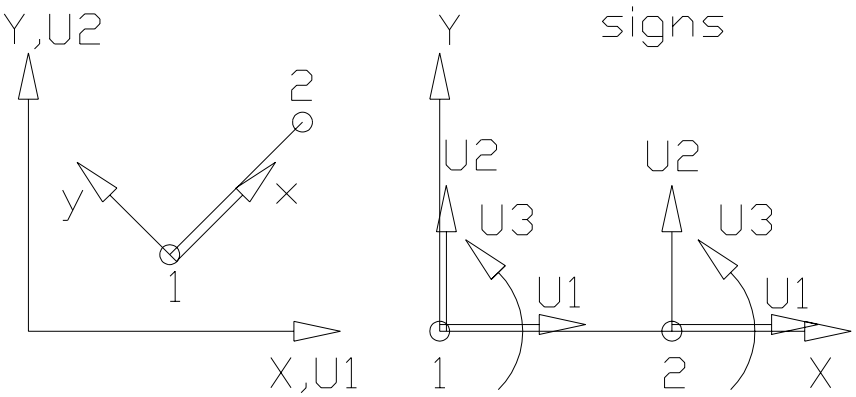
Plane Stress Isoparametric Element No. 11

- Cubic Isoparametric Serendipity element
- Quality of displacements excellent
- Quality of stresses in the Gauss- points excellent
- Quality of stresses in the corner nodes good
- Computing effort: Very high
- Size of element stiffness matrix: 24 * 24



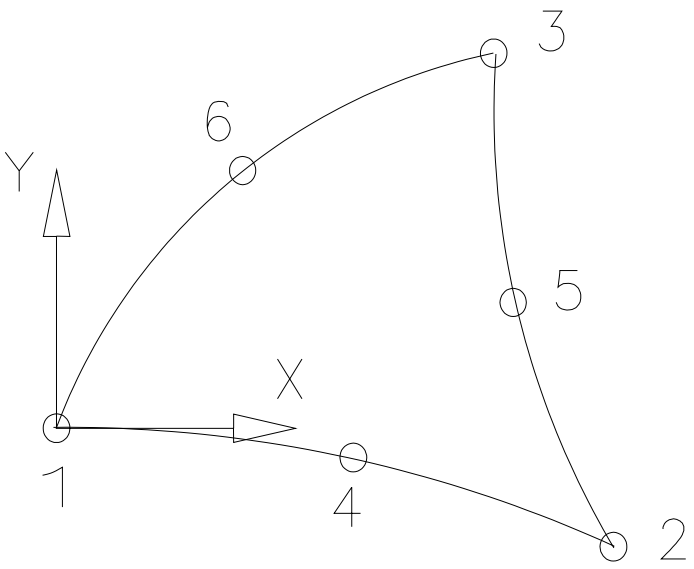
Beam No. 13

- Linear function for tensile stress, cubic function for bending stress
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 8 * 8



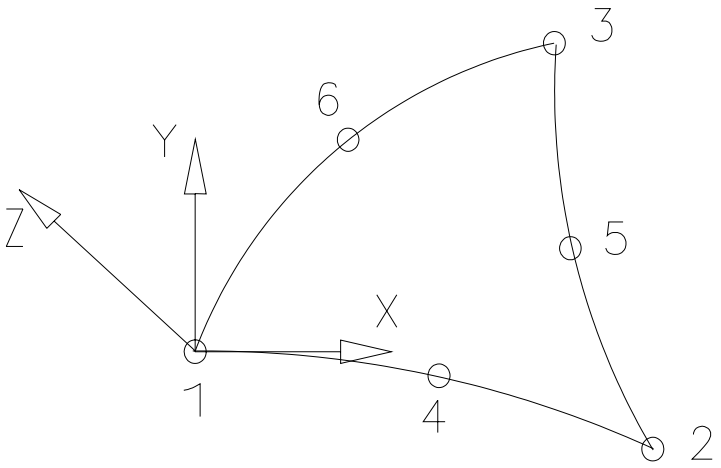
Plane Stress Isoparametric Element No. 14

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Quality of stresses in the Gauss- points very good
- Quality of stresses in the corner nodes good
- Computing effort: medium
- Size of element stiffness matrix: 12 * 12



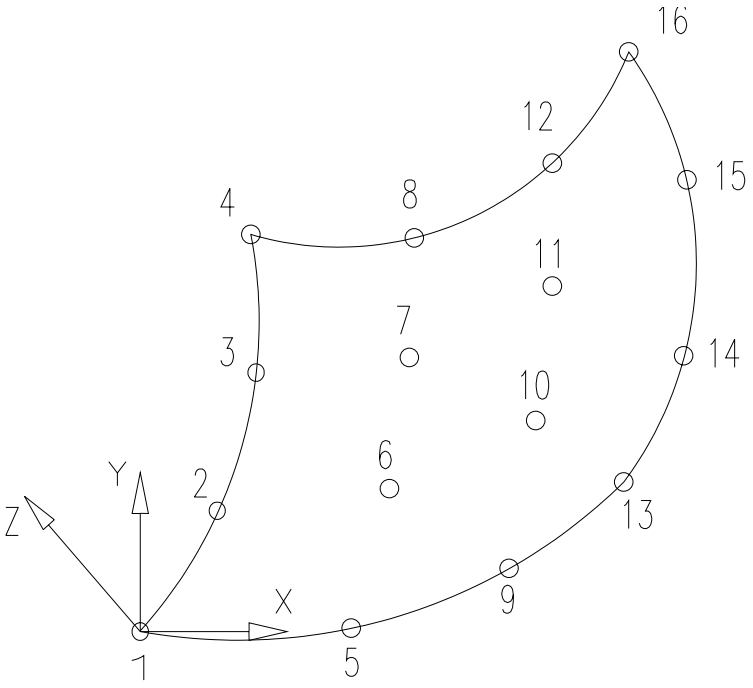
Isoparametric Plate Element No. 18

- Quadratic Isoparametric Serendipity element following Reissner- Mindlin's theory
- Quality of displacements very good
- Quality of stresses in the Gauss- points good
- Quality of stresses in the corner nodes acceptable
- Computing effort: medium
- Size of element stiffness matrix: 18 * 18



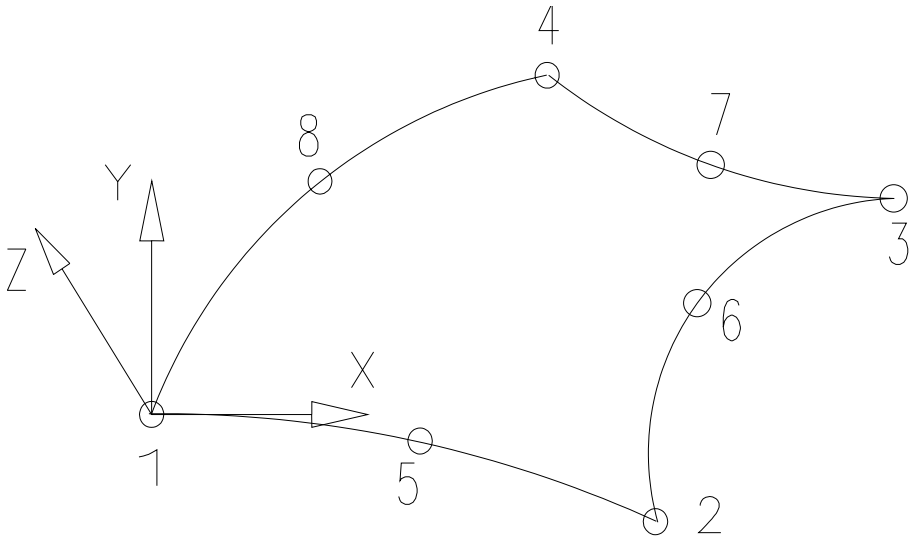
Isoparametric Plate Element No. 19

- Cubic Isoparametric Lagrange element following Reissner- Mindlin's theory
- Quality of displacements very good
- Quality of stresses in the Gauss- points very good
- Quality of stresses in the corner nodes good
- Computing effort: High
- Size of element stiffness matrix: 48 * 48



Isoparametric Plate Element No. 20

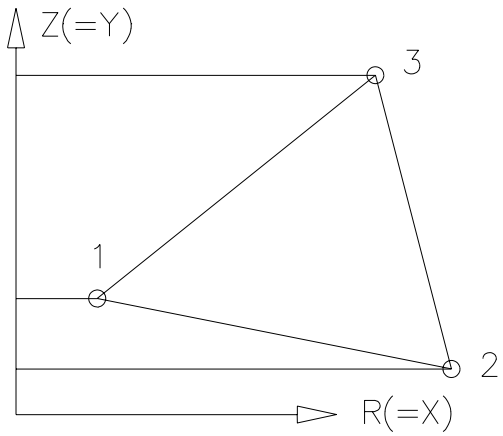
- Quadratic Isoparametric Serendipity element following Reissner- Mindlin's theory
- Quality of displacements very good
- Quality of stresses in the Gauss- points good
- Quality of stresses in the corner nodes quite good
- Computing effort: medium
- Size of element stiffness matrix: 24 * 24



Axisymmetric problems:

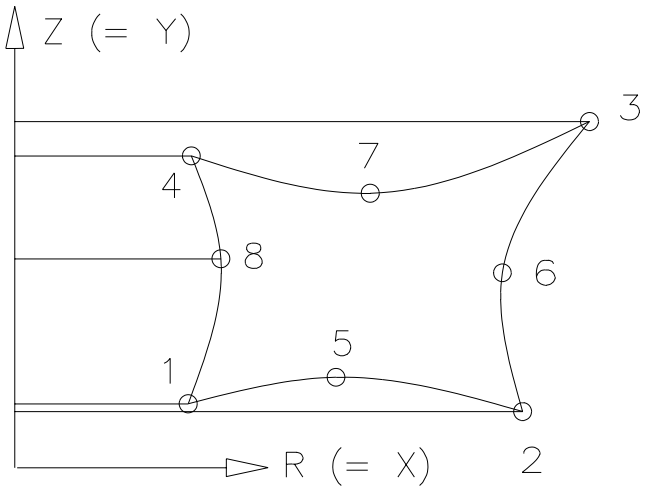
Torus No. 6

- Linear function
- Quality of displacements average
- Quality of stresses in the corner nodes inaccurate
- Computing effort: Low
- Size of element stiffness matrix: 6 * 6



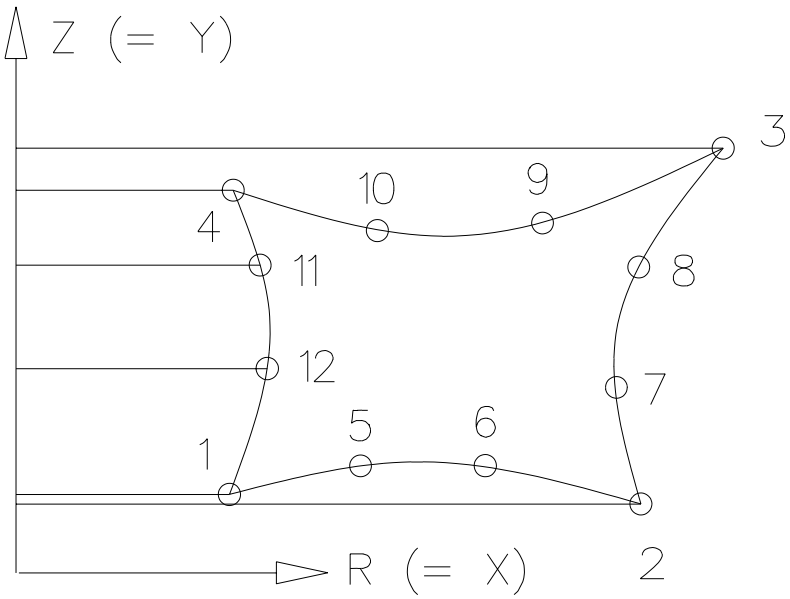
Torus No. 8

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Quality of stresses in the Gauss- points very good
- Quality of stresses in the corner nodes good
- Computing effort: High
- Size of element stiffness matrix: 16 * 16



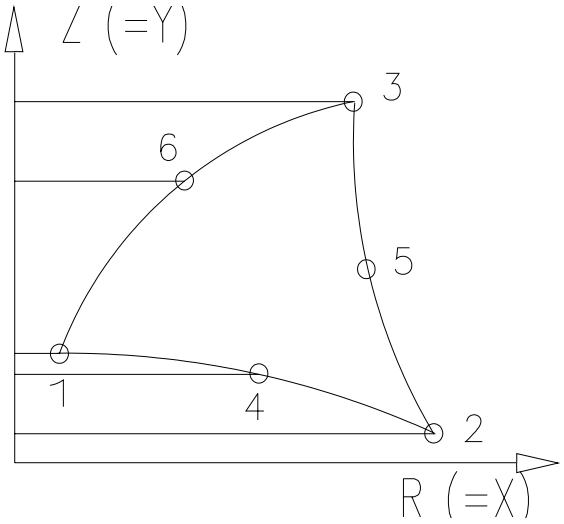
Torus No. 12

- Cubic Isoparametric Serendipity element
- Quality of displacements excellent
- Quality of stresses in the Gauss- points excellent
- Quality of stresses in the corner nodes good
- Computing effort: Very high
- Size of element stiffness matrix: 24 * 24



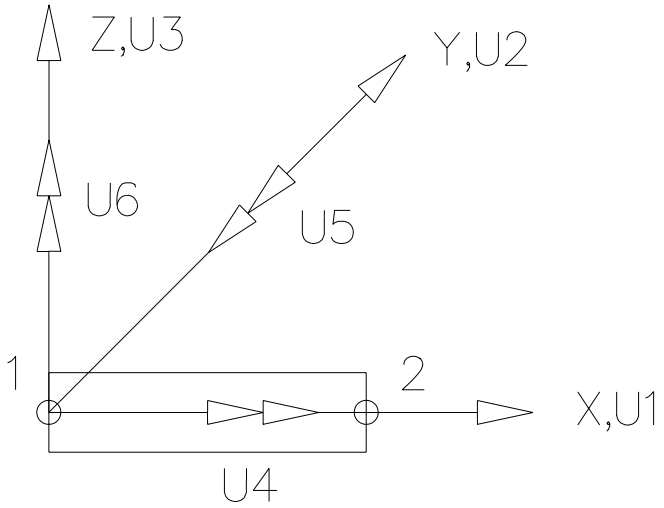
Torus No. 15

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Quality of stresses in the Gauss- points very good
- Quality of stresses in the corner nodes good
- Computing effort: High
- Size of element stiffness matrix: 12 * 12



Cam No. 5

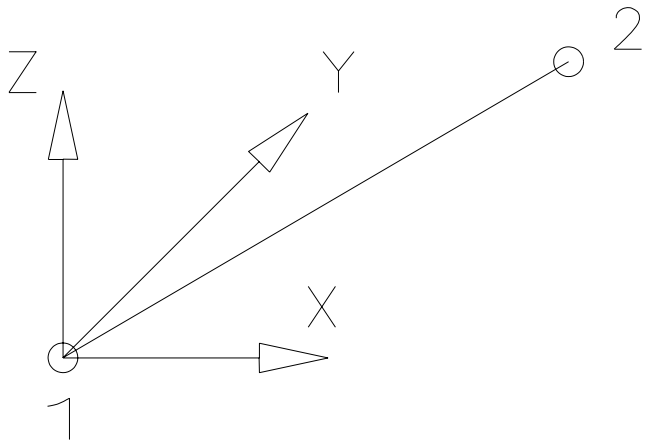
- Linear function for torsion and tensile stress, cubic function for bending stress
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 12 * 12



Space problems:

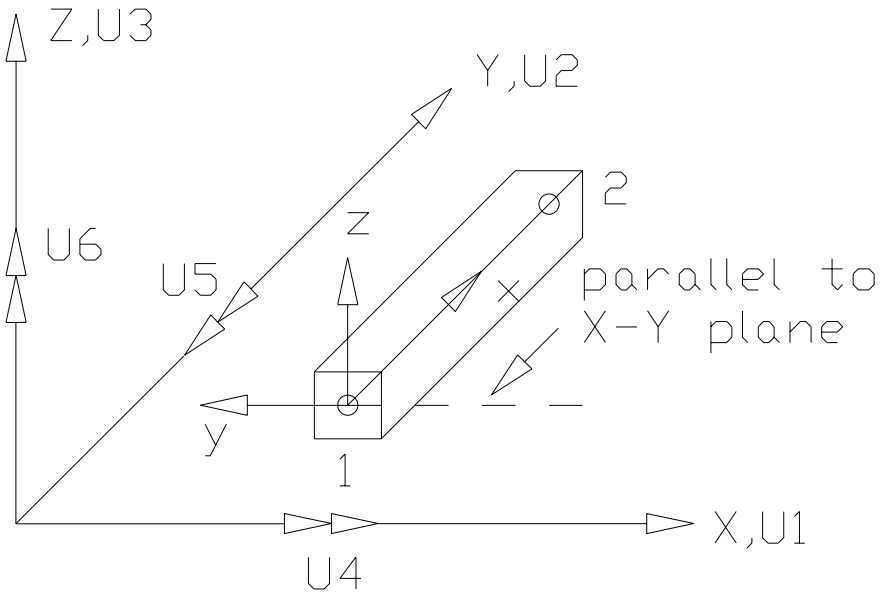
Truss No. 4

- Linear function
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Minimal
- Size of element stiffness matrix: 6 * 6



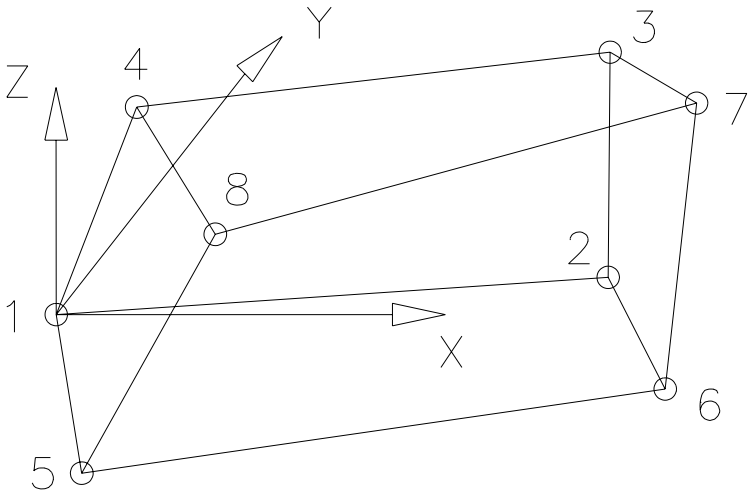
Beam No. 2

- Linear function for tensile stress, cubic function for bending stress
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 12 * 12



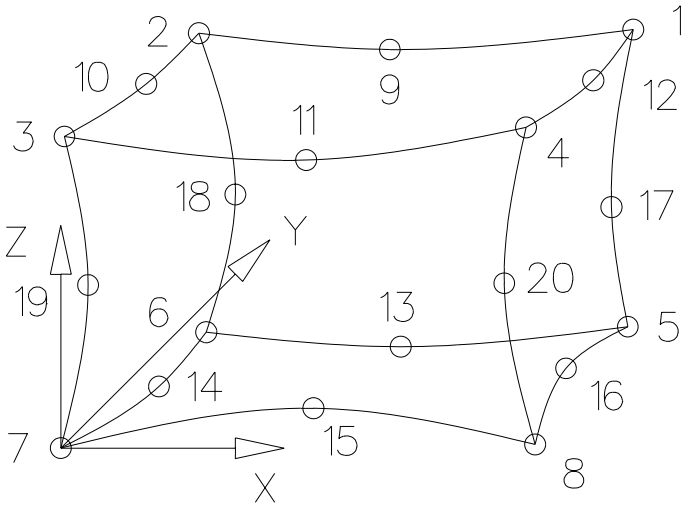
Hexahedron No. 1

- Linear shape functions
- Quality of displacements average
- Stresses in the Gauss- points useable
- Stresses in corner nodes inaccurate
- Computing effort: very high
- Size of element stiffness matrix: 24 * 24



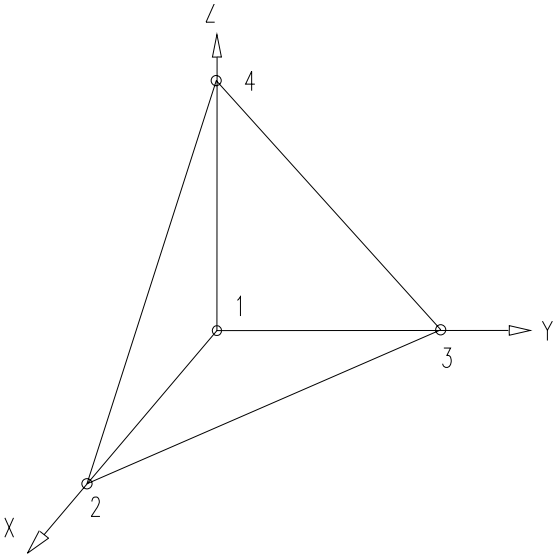
Hexahedron No. 10

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Stresses in the Gauss- points very good
- Stresses in corner nodes good
- Computing effort: extremely high
- Size of element stiffness matrix: 60 * 60



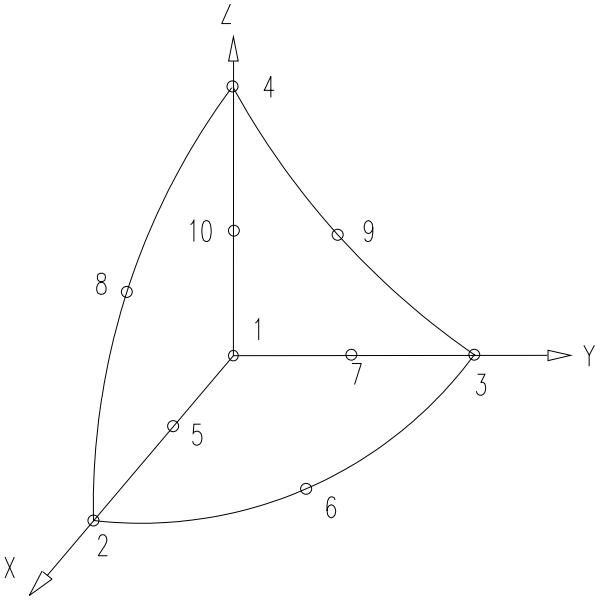
Tetrahedron No. 17

- Linear shape functions
- Quality of displacements bad
- Stresses in the Gauss- points inaccurate
- Stresses in corner nodes very inaccurate
- Computing effort: medium
- Size of element stiffness matrix: 12 * 12



Tetrahedron No. 16

- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Stresses in the Gauss- points very good
- Stresses in corner nodes good
- Computing effort: very high
- Size of element stiffness matrix: 30 * 30



The Z88 computing units:

Overview:

Z88 always exclusively works at the tasks required at the moment. Thus, Z88 is no gigantic, monolithic program, but consists of several separate running modules according to the UNIX philosophy "Small Is Beautiful". They are loaded into the main memory according to your requirements, execute their tasks and release the main memory again. In this way Z88's achieves its enormous speed and faultlessness beating many other FE programs! The Z88 modules communicate by files, cf. Chapter 3.

Short description of the modules:

I. The Solver

The **solver** is the heart of any FEA system. It reads the general structure data Z88I1.TXT and the boundary conditions Z88I2.TXT. Basically, the Z88 input files can be created by CAD converter Z88X, by COSMOS converter Z88G, by net (or mesh) generator Z88N, by editor or word processor system or by a mixed procedure, e.g. by CAD and editor. The solver generates prepared structure data Z88O0.TXT and processed boundary conditions Z88O1.TXT, calculates the element stiffness matrices, compiles the total stiffness matrix, scales the system of equations, solves the (huge) system of equations and stores the displacements in Z88O2.TXT. Therefore, the main task of every FEA system, the calculation of displacements, is solved. Thereupon, if you wish, the stresses can be calculated by Z88D and/or nodal forces by Z88E.

Z88 features two totally different solvers:

Z88F: This is a so-called direct solver with skyline storing scheme and an in-situ Cholesky solver. It is the standard solver of Z88, easy to handle and very fast for small and medium structures. However, like any direct solver Z88F reacts badly on ill- numbered nodes but you may improve the situation with the Cuthill- McKee program Z88H. Z88F is your choice for small and medium structures, up to 20,000 ... 50,000 degrees of freedom.

Z88I1 and **Z88I2**: This is a so-called iteration solver featuring two modules. Z88I1 computes the pointers for the storage scheme of the total stiffness matrix. Z88I2 computes the stiffness matrices, addes the boundary conditions and solves the system of equations by the method of conjugate gradients featuring SOR- preconditioning or precontitioning by an incomplete Cholesky decomposition depending on your choice. Like any iteration solver Z88I1/Z88I2 deals well with bad node numbering, a run with the Cuthill- McKee program Z88H may improve the situation further, however. Z88I1/Z88I2 is your choice for large structures.

II. The link to CAD programs

The **CAD converter Z88X** converts DXF files from CAD systems into Z88 input files (net generator input file Z88NI.TXT, general structure data Z88I1.TXT, boundary conditions Z88I2.TXT and stress parameters file Z88I3.TXT) or, and this is the real goodie, also converts Z88 input files into DXF files. You cannot only produce input data in the CAD system and then use in Z88, but you can also complete Z88 entry files which are always

simple ASCII files, e.g. by text editor, by word processing, by EXCEL or e.g. by your own special programs and then convert the data sets back into the CAD system by CAD converters Z88X. In the CAD system you can add more informations, then push the data again to Z88. This flexibility is unique!

The new **COSMOS converter Z88G** reads FEA input files following the COSMOS or the NASTRAN format and generates the Z88 input files Z88I1.TXT, Z88I2.TXT and Z88I3.TXT automatically. You may produce COSMOS or NASTRAN data files by various CAD programs. However, Z88G is properly tested with Pro/ENGINEER with the Pro/MECHANICA option by Parametric Technology, USA. Thus, you may directly use Pro/ENGINEER 3D models with Z88 !

The new **Cuthill- McKee program Z88H** was mainly designed for use with Z88G. It allows the re-numbering of finite elements meshes and may heavily decrease the memory needs for meshes generated by automeshers i.e. Pro/MECHANICA.

III. The net generator for ordered meshes

The **net (or mesh) generator Z88N** reads the super structure data Z88NI.TXT and computes the general structure data Z88I1.TXT. In principle, the net generator file Z88NI.TXT has the same construction as the file of the general structure data Z88I1.TXT. It can also be generated by CAD converters Z88X, by editor or word processor system or with a mixed procedure.

IV. The postprocessors

Stresses are calculated by **Z88D**. Z88F or Z88I1 and Z88I2 must have run before. Z88D reads a small parameter file Z88I3.TXT and stores the stresses in Z88O3.TXT.

Nodal forces are calculated by **Z88E**. Z88F or Z88I1 and Z88I2 must have run before. Z88E stores the nodal forces in Z88O4.TXT.

The **plot programs Z88P** and **Z88O** plot deflections and stresses on the CTR, Z88P also on a HP-GL plotter or a printer capable of HP-GL, e.g. HP LaserJet. Z88O and Z88P are suitable for a quick inspection of the undeflected and the deflected structures as well as for showing the stresses. Of course, you can show undeflected structures on your CAD program capable of DXF via CAD converter Z88X, too, but Z88O and Z88P are much faster.

V. The file checker

The **file checker Z88V** checks the input files Z88NI.TXT or Z88I1.TXT to Z88I3.TXT for formal correctness. In addition, it can show the actual memory defined by you in the file Z88.DYN.

All modules of Z88 request Memory dynamically:

The user can define this in the file Z88.DYN. Z88 is delivered with default values which you can and also should change if necessary. This is possible at any time. The Z88 modules are genuine 32 bit (or 64 bit) programs and request their memory by operating system calls via *calloc*. The header file Z88.DYN provides how much memory shall be requested. You can request all virtual memory (virtual memory = main memory + swap area), which is provided by the operating system. **Therefore there is no limit for the size of the Z88 finite element**

structures ! You can also fix the language for Z88 in the file Z88.DYN: keywords *ENGLISH* or *GERMAN*.

Multitasking of Z88:

Absolute multitasking is possible under Windows and UNIX, i. e. several Z88 modules or other genuine Windows programs can run parallel. Make sure that you do not overlap the windows (put them side by side), as if the Z88 modules have once started they are not evaluating WM_PAINT signals for speed reasons. This means, that, although the Z88 programs are properly working, displays and window images can be destroyed if you enlarge, reduce, move or cover Z88 windows by other programs. This does not have any influence on the computing results and only by this trick the outstanding speed of Z88 can be gained. Keep in mind that big space structures, e.g. with 20 nodes hexahedrons, can put very heavy load on your computer which can slow down the machine totally. Thus, let Z88 run alone and do not start any memory eaters like the various office programs.

Hints for the start of Z88:

Windows:

All Z88 modules can be started directly via Explorer, from a group which contains the various Z88 modules or via *Start* > *Run*. It suffices to call the Z88-Commander Z88COM for launching all other modules.

UNIX:

Launch the modules directly from a UNIX shell, from the Z88-Commander Z88COM, or, as an extended possibility, e.g. for large-caliber night runs, from a shell-script (*sh, bash, ksh* etc.). **You have all unlimited liberties of the UNIX operating system.** All modules except Z88COM and Z88P can be started in text mode from consoles, but naturally also in an X window. As Motif programs the Z88-Commander Z88COM and the plot programs Z88O and Z88P are to start from an X-term.

For a convenient use of Z88, fire up your X-Window-manager, open an X-term and launch Z88COM. Put Z88COM and the X-Term, which started Z88COM, side-by-side or over-and-under to see both.

The Input and Output of Z88:

The input and output files are generated either by an editor (e.g. the *editor* or *notepad* of Windows, DOS editors like *edit*, UNIX tools like *vi, emacs, joe*), word processor program (e.g. *WinWord* etc.), spreadsheet program (e.g. *Excel*) or via CAD converter Z88X directly in a CAD program, which can read and write DXF files (e.g. *AutoCAD*). Or import COSMOS or NASTRAN data files produced by *Pro/ENGINEER* into Z88 by use of the COSMOS converter Z88G.

For the user this means maximum flexibility and transparency, as the input and output files of Z88 are quite simple ASCII text files. You can fill the files by arbitrary tools or by hand, and also by self-written programs, of course. Only make sure to meet the Z88 conventions for the respective file structure cf. Chapter 3.

You can modify output files as you like, enlarge them with your own comments, reduce them to the essential or use them as input for other programs.

Dimensions, i. e. measurement units, are not used explicitly. You can work in optional measurement systems, e.g. in the Metric or Imperial measurement system. Use inches, Newtons, pounds, tons, millimeters, meters, yards - whatever you prefer. But make sure to keep the one chosen measurement units throughout all computations of this structure. Example: You want to work with mm and N so Young's modulus must be used in N/mm*mm.

Note:

The Z88 input files read always:

+ Z88G.COS COSMOS data file from 3D-CAD system for COSMOS converter Z88G
+ Z88G.NAS NASTRAN data file from 3D-CAD system for COSMOS converter Z88G
+ Z88X.DXF Exchange file for CAD programs and for CAD converter Z88X
+ Z88NI.TXT Input file for the net generator of Z88N
+ Z88I1.TXT Input file (general structure data) for the FE processor of Z88F
+ Z88I2.TXT Input file (boundary conditions) for the FE processor of Z88F
+ Z88I3.TXT Input file (control values) for the stress processor of Z88D
+ Z88I4.TXT Input file (control values) for the iteration solver Z88I1/Z88I2

The Z88 output files read always

+ Z88O0.TXT Prepared structure data for documentation purposes
+ Z88O1.TXT Prepared boundary conditions for documentation purposes
+ Z88O2.TXT Computed displacements
+ Z88O3.TXT Computed stresses
+ Z88O4.TXT Computed nodal forces

These file names are expected from the Z88 modules and they must reside in the same Directory as the Z88 modules. You cannot allocate your own names for data sets. Of course, you may rename the Z88*.* files after all calculations have been done and save them in other directories.

Making:

You may allways create the net generator file Z88NI.TXT, the general structure data file Z88I1.TXT, the boundary conditions file Z88I2.TXT and the control values file Z88I3.TXT for the stress prozessor by hand using an editor or the like.

Using automatic generation consider the following possibilities:

CAD system, e.g.	*creates*	*converter*	*creates*	*net generator*	*creates*
Pro/ENGINEER (with Pro/ME-CHANICA)	Z88G.COS Z88G.NAS	**Z88G**	Z88I1.TXT, Z88I2.TXT, Z88I3.TXT	not necessary	files still exist
AutoCAD	Z88X.DXF	**Z88X**	Z88NI.TXT	**Z88N**	Z88I1.TXT
AutoCAD	Z88X.DXF	**Z88X**	Z88I1.TXT, Z88I2.TXT, Z88I3.TXT	not necessary	files still exist

Z88 protocol files:

The Z88 modules always store protocol files .LOG, e.g. Z88F.LOG documents the steps or errors of the calculation of Z88F. Look at the various .LOG files in case of doubt. They also document the current memory needs. **UNIX:** If different users work in the same Z88 directory, make sure to have the proper permissions for the .LOG files, too. Use *umask*.

Printing of Z88 files

Is not supported by the Z88- Commanders. You print them by the Explorer of Windows or by an editor or word processing program. Use the printing routines of the UNIX operating system.

Which Z88 finite Element types can be produced automatically ?

element type	*function*	*COSMOS NASTRAN (Z88G)*	*DXF (Z88X)*	*super element (Z88N)*	*creates FE (Z88N)*
Hexahedron No.1	linear	No	Yes	No	-
Hexahedron No.10	quadratic	No	Yes	Yes	Hexa No.10 & No.1
Tetrahedron No.16	quadratic	Yes	No	No	-
Tetrahedron No.17	linear	Yes	No	No	-
Plane stress No.3	quadratic	No	Yes	No	-
Plane stress No.7	quadratic	Yes	Yes	Yes	Plane stress No.7
Plane stress No.11	cubic	No	Yes	Yes	Plane stress No.7
Plane stress No.14	quadratic	Yes	Yes	No	-
Torus No.6	linear	No	Yes	No	-
Torus No.8	quadratic	Yes	Yes	Yes	Torus No.8
Torus No.12	cubic	No	Yes	Yes	Torus No.8
Torus No.15	quadratic	Yes	Yes	No	-
Plate No.18	quadratic	Yes	Yes	No	-
Plate No.19	cubic	No	Yes	No	-
Plate No.20	quadratic	Yes	Yes	Yes	Plate No.19 & No.20
Truss No.4	exact	No	Yes	No	-
Truss No.9	exact	No	Yes	No	-
Beam No.2	exact	No	Yes	No	-
Cam No.5	exact	No	Yes	No	-
Beam No.13	exact	No	Yes	No	-

All Z88 files:

Name	Type	Direction	Purpose	change, modify	MS-Win	UNIX
Z88.DYN	ASCII	Input	Memory & Language header file	Yes,Recom.	Yes	Yes
Z88G.COS	ASCII	Input	COSMOS to Z88	Yes, 1)	Yes	Yes
Z88G.NAS	ASCII	Input	NASTRAN to Z88	Yes, 1)	Yes	Yes
Z88X.DXF	ASCII	In/Output	DXF from and to Z88	Yes, 1)	Yes	Yes
Z88NI.TXT	ASCII	Input	net generator input file	Yes	Yes	Yes
Z88I1.TXT	ASCII	Input	general structure data	Yes	Yes	Yes
Z88I2.TXT	ASCII	Input	constraints	Yes	Yes	Yes
Z88I3.TXT	ASCII	Input	stress parameter header file	Yes	Yes	Yes
Z88I4.TXT	ASCII	Input	header file for iteration solver	Yes	Yes	Yes
Z88O0.TXT	ASCII	Output	processed structure data	Possible	Yes	Yes
Z88O1.TXT	ASCII	Output	processed constraints	Possible	Yes	Yes
Z88O2.TXT	ASCII	Output	computed displacements	Possible	Yes	Yes
Z88O3.TXT	ASCII	Output	computed stresses	Possible	Yes	Yes
Z88O4.TXT	ASCII	Output	computed nodal forces	Possible	Yes	Yes
Z88O5.TXT	ASCII	Output	for internal use of Z88P	No 2)	Yes	Yes
Z88O6.TXT	ASCII	Output	Main HP- GL file from Z88P	Yes 1)	Yes	Yes
Z88O7.TXT	ASCII	Output	Aux. HP- GL file from Z88P	Yes 1)	Yes	Yes
Z88O8.TXT	ASCII	Output	for internal use of Z88O	No 2)	Yes	Yes
Z88P.COL	ASCII	Input	Color header file Z88P MS-Win	Possible	Yes	No
Z88O.OGL	ASCII	Input	Color header file Z88O MS-Win	Possible	Yes	No
Z88.FCD	ASCII	Input	Fonts, Colors, Dimens. UNIX for Z88COM, Z88O and Z88P	Possible	No	Yes
Z88COM.CFG	ASCII	Input	configuration file Z88COM	No 2)	Yes	No
Z88O1.BNY	Binary	In/Output	fast communication file	No 3)	Yes	Yes
Z88O2.BNY	Binary	In/Output	fast communication file	No 3)	Yes	Yes
Z88O3.BNY	Binary	In/Output	fast communication file	No 3)	Yes	Yes
Z88O4.BNY	Binary	In/Output	fast communication file	No 3) 4)	Yes	Yes

(1) in principle yes, but not necessary, automatically produced
(2) only if needed
(3) positively no, otherwise serious faults
(4) may become quite large, for communication of iterations solver Z88I1 and Z88I2

1.2 HOW TO INSTALL Z88 FOR WINDOWS 95 TO XP

Remark: We could of course use the standard installation routines or ready-to-run installation tools for Z88, but as there are no hidden .DLL files, no .INI files are to be modified and no subdirectories are created, we leave it alone. You will see, Z88 installs quite simply:

Windows in five steps:

1st step: Copy the Z88 files into a new or existing directory:

We assume that you have copied the file **Z88RUNE.EXE** from the Z88 diskette into a new directory named Z88 on hard disk D: . If you have copied Z88 to C:\SOMEWHERE then replace D:\Z88 in the following description against C:\SOMEWHERE. Now launch **Z88RUNE.EXE**, e.g. by Start > Run oder from the "DOS prompt". This uncompresses Z88. No other modifications are made and the Windows system files are not modified.

Now you may delete Z88RUNE.EXE to prohibit another start resulting in overwriting your own input files.

2nd step: Make Z88 ready to run:

Two different methods are usual under Windows:

(1) Folder on the desktop:
Define a new folder on the desktop: Point to a free area on the desktop, press right mouse key, *New > Folder*. Name the new folder e.g. *Z88*. Include at least Z88COM into the new folder: Open folder by double click, *File > New > Shortcut*. Enter: *D:\Z88\Z88COM.EXE*, *Next > Z88COM* and *Finish*
With the same procedure you can add the other Z88 modules (*File > New > Shortcut*): Z88F, Z88I1, Z88I2, Z88D, Z88E, Z88G, Z88H, Z88X, Z88N, Z88V, Z88O, Z88P. However, skip it, if you want to launch the modules exclusively by the Z88 Commander Z88COM.

(2) Installation in "Start":
Point at the *Start* button, press right mouse button, select *Open*. Open folder *programs* by double click. *File > New > Shortcut*, enter for command line: D:\Z88\Z88COM.EXE, *Next* > , name the icon e.g. *Z88*, *Finish*. You can also place a whole folder here.

3rd step: Enter your favourite editor in Z88

You may produce all input files either by a CAD program which can read and generate DXF files in cooperation with the CAD converter Z88X or also write by editor since Z88 operates with ASCII files, however. An editor for looking at the Z88 results is also very useful. So you should define it:

Suitable editors are under Windows *edit* or *editor* from *Start > Programs > Accessories*. However, you can also use old-fashioned DOS programs like *WordStar* or the *Norton Editor*.

Assume you want to use *edit* (included in Windows) as Z88 editor . Start Z88COM, *File > Define Editor*, enter in textfield *Editor Name* any text, e.g. EDIT or TRUE-LIES. In textfield *Editor Call, if nes. Path* enter the program name, here *edit*. You want to work with Notepad:

enter in textfield *Editor Name* any text, e.g. MY-NOTEPAD, enter in textfield *Editor Call, if nes. Path*. the program name *notepad*. Further example: Word for Windows. You must find out where Word for Windows is located. Go ahead: *Start > Find > Files or Folders* : *winword.exe*. Let's assume WinWord is located in C:\MSOffice\Winword. Thus you could enter in Z88COM: *Word4Windows* and *C:\MSOffice\Winword\winword* . **Make sure when using Winword that you work and save in plain text mode!**

4th step: Add an Internet Browser for Z88's OnLine:

Integrate your favourite Internet Browser into Z88. This may be *Mozilla, Mosaic* or *MS Internet Explorer*.

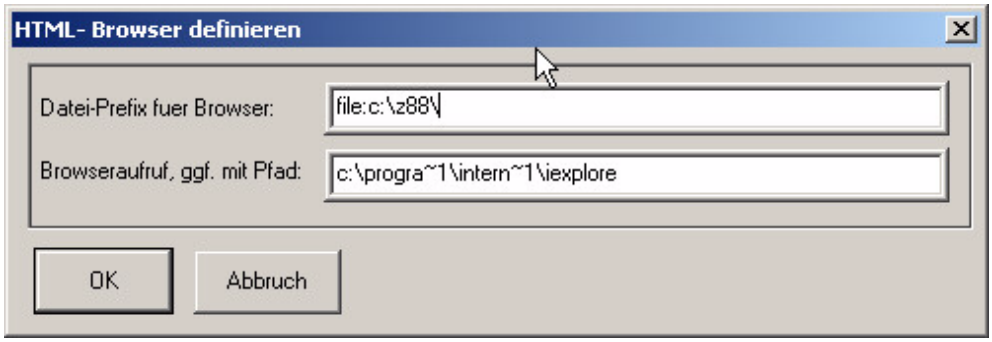
(1) the next step is very important: Z88 must be able to start the Browser ! Either you must put it into the PATH or enter the PATH in Z88COM or copy the whole Browser into the Z88 directory.

State at first where your Internet-Browser is located. Use *Start > Find > Files or Folders*. The Microsoft Internet-Explorer is called *iexplore.exe*, Mozilla is called *mozilla.exe*. Note down the found path.

1st possibility: Type in path into the PATH variable: *Start > Settings > Control Panel > System > Advanced > Environment*. You should always do this if the path also includes blanks.
Example: The Internet-Explorer is located in Windows 2000: *c:\Program Files\Internet Explorer* (with blanks between Program and Files and Internet and Explorer !).

Do we assume your previous PATH variable looks as follows:
D:\WATCOM\BINNT;D:\WATCOM\BINW;
With the Browser included PATH should look like:
D:\WATCOM\BINNT;D:\WATCOM\BINW; c:\Program Files\Internet Explorer;
Logoff and login again.

2nd possibility: Enter path in Z88COM directly: Assume Mozilla is located in D:\MOZILLA\PROGRAM\. Start Z88COM, *File > Define Browser* and enter in textfield *Browser Call, if nes. Path*:
D:\MOZILLA\PROGRAM\MOZILLA
Now for advanced users: The Internet-Explorer is located in Windows 2000: *c:\Program Files\Internet Explorer*. Because of the blanks between Program and Files and Internet and Explorer you may proceed as follows: Start Z88COM, *File > Define Browser* and enter in textfield *Browser Call, if nes. Path*:
c:\progra~1\intern~1\iexplore



This are the "old" DOS names with a maximum of 8 letters without any blanks. How do you

get such old- fashioned DOS names? Try **dir /X** in a DOS windows (Command Prompt). It works, really.

(2) Take into account that most Internet-Browsers immediatelly try to contact the Internet. Now they are to load a local HTML file. Thus, various file prefixes depending on the used Browser must be fixed. For Mozilla the prefix is *file:///Z88 path*, e.g. *file:///D:/Z88/* for Microsoft Internet Explorer the prefix is *file:Z88 path*, e.g. *file:D:\Z88* .

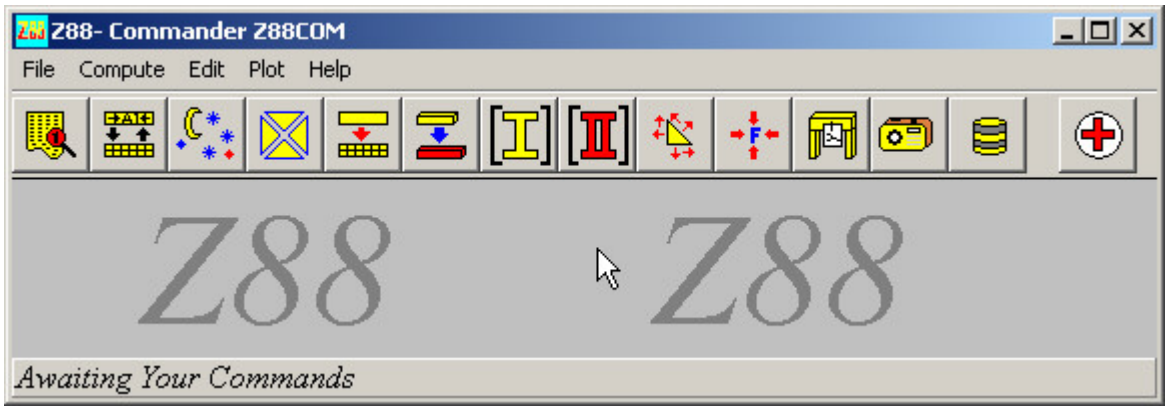
With above explanations and assuming that you have copied all Z88 files to D:\Z88, the Browser definitions in Z88COM would read:

Microsoft- Internet Explorer:	**Mozilla** (also true for Netscape):
File Prefix for Browser: *file:d:\z88*	File Prefix for Browser: *file:///d:/z88/*
Browser Call, if nes. Path : *c:\progra~1\intern~1\iexplore*	Browser Call, if nes. Path: *d:\progra~1\mozilla.org\Mozilla\Mozilla.exe*

5th step: Launch Z88:

Z88 is ready to run. You may fire away immediatelly by launching the Z88- commander Z88COM and using the OnLine help system. Proceed with example 5.1.

Notes for the Z88-Commander Z88COM



It starts all Z88 modules, provided that you don't want to start them stand-alone (which is possible any time and without any restrictions), permits the immediate editing of all input and output files and calls the context sensitive online-help. So you launch the online- help: Select in an arbitrary pulldown menu the point *Help Mode*. The cursor changes to a question mark. If you click now on a menu item the menu point is not executed but the associated help appears. The help mode keeps active until you click on a menu item *Help Mode* again.

Furthermore, the Z88-Commander Z88COM offers support for HP-GL files, such as generating header sequences for HP plotters and HP LaserJets.

Z88COM files your entries for the Internet-Browser and editor in a file Z88COM.CFG. If this file should be destroyed accidentally, you can edit Z88COM.CFG by hand:
1st line: Editor name
2nd line: Editor call
3rd line: Browser prefix
4th line: Browser call
Example:

Word4Windows
C:\MSOffice\Winword\Winword
File:///D:/z88/
d:\progra~1\mozilla.org\Mozilla\Mozilla.exe

... And how do you remove Z88 ?

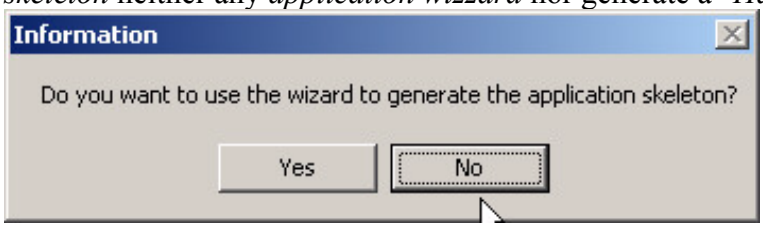
Simply delete all files in the directory containing Z88. Then delete the directory if necessary. You should delete the links we made for Windows in chapter 1.2. That's all !

And how to compile Z88 for Windows?

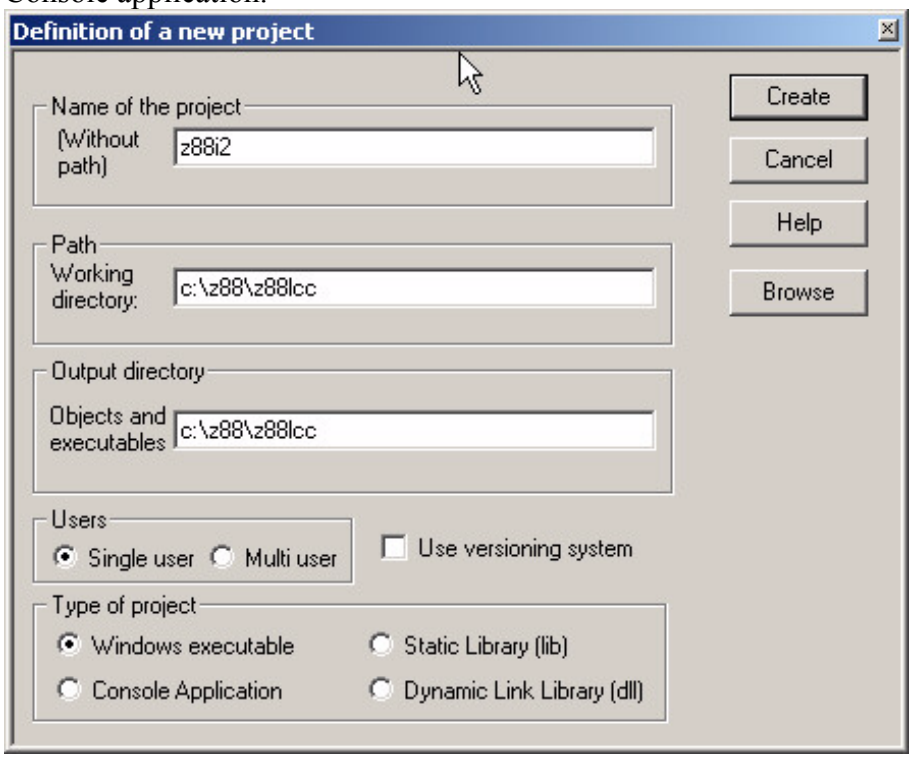
Only if you want to add improvements to Z88 you will need to compile the package. Every Windows C or C++ compiler should work properly. I tried the free LCC and the compilers from Microsoft (Visual C++), Borland (C++ Builder) and Watcom (C/C++ 10.6 and OpenWatcom). Because every brand uses its own project management we can't use ready- to-go makefiles. How do we proceed? I prepared for you a compiler session for the iterations solver part 2 Z88I2 featuring the free LCC, but Microsoft Visual C++, the Borland C++ Builder and the Watcom C/C++ compiler sessions are very similar :

1. Launch a new project.

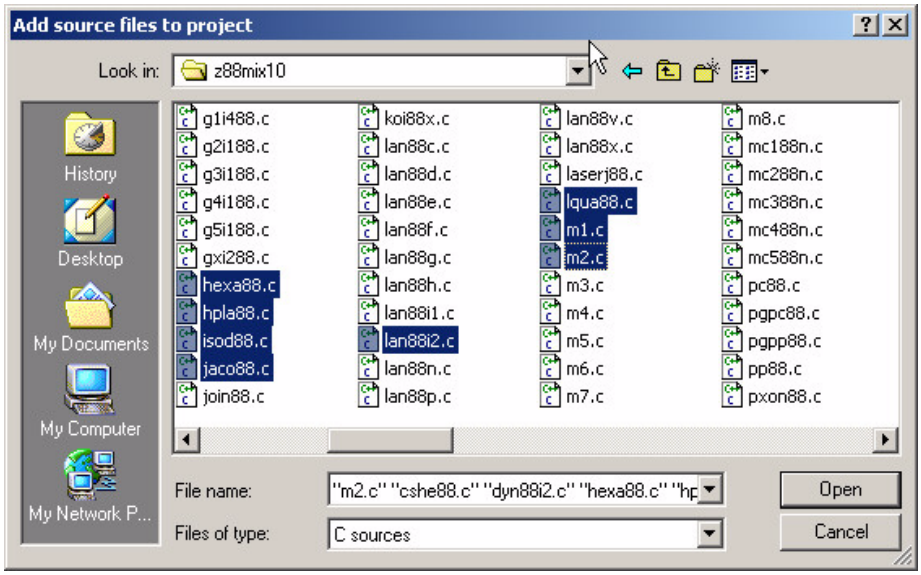
2. Be sure to choose a pure Win32 application "without anything" . Don't use any *application skeleton* neither any *application wizzard* nor generate a *"Hallo World" application*.



3. Enter your favorite directories and make sure to generate a Windows application not a Console application.



4. Add the matching sources to your project (ref. table below), i.e. the C- sources and the appropriate Ressource file **.rc* (in this case Z88I2.RC).



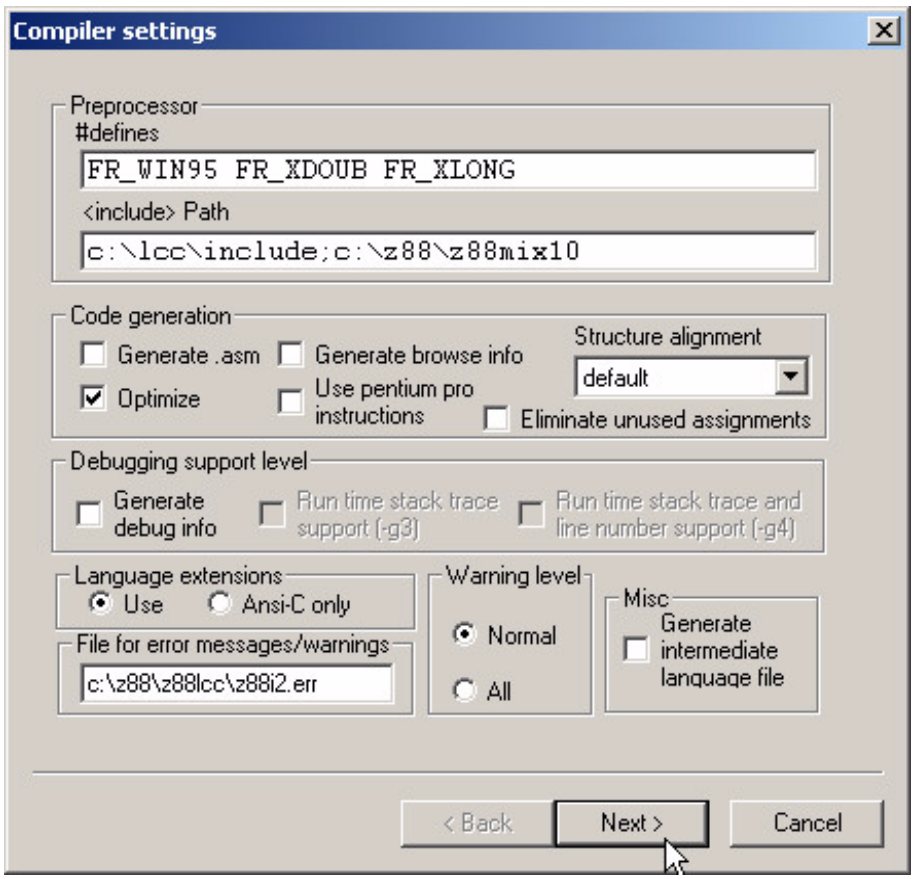
5. a most important step:

Adjust the compiler, the linker and the ressource compiler: Tell your compiler system
- where the *Header files Z88*.H* (the so-called include files) are located
- which *Defines* are necessary (here *FR_WIN95, FR_XDOUB and FR_XLONG*) (ref. Table below).

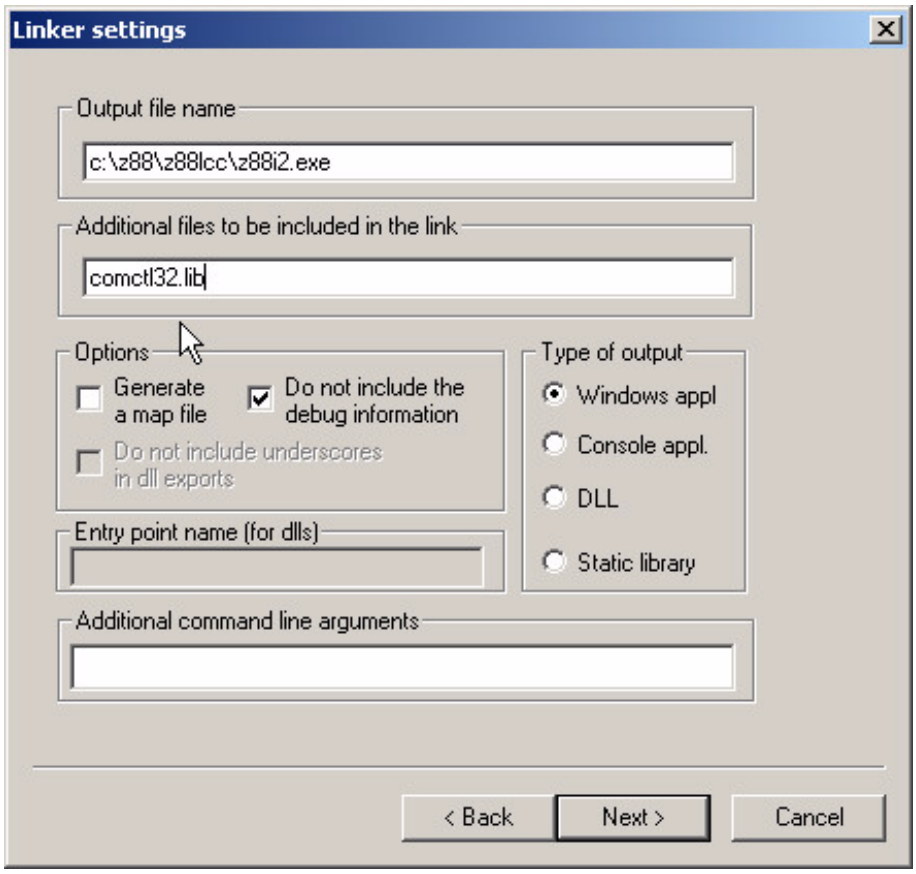
Most of the compiler malfunctions will result from wrong paths and missing defines!

You may leave the Debug infomations off. Choose a soft to medium optimization level.

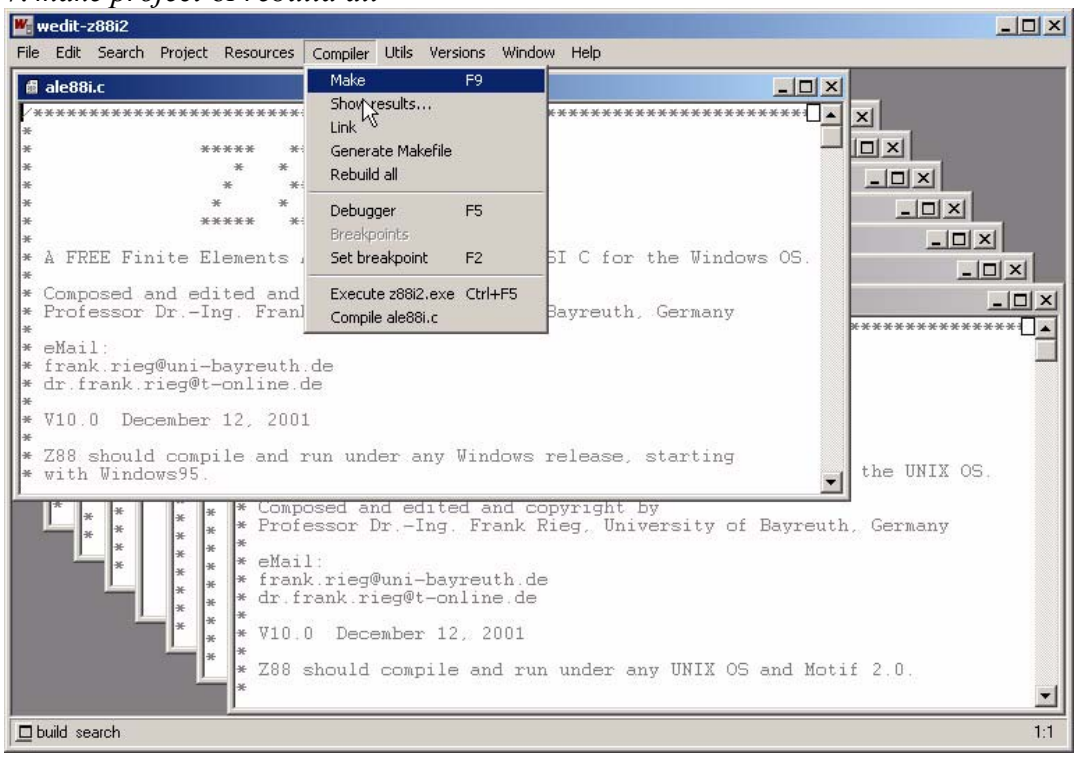
If in doubt skip optimization if you don't know the details. If you have never heard about a *Framepointer* or *inline functions*, *Parameter passing by Stack* or *Parameter passing by Memory* and *Aliases*, then keep your hands off! (if you are interested in such stuff, you may consult the book *Rieg,F.; Hackenschmidt, R.: Softwaretechnik für Ingenieure, Hanser Verlag, München Wien 2001* (in German Language)).



6. Don't forget to link against the library *comctl32.lib* (Common Controll Library) ! It is part of your compiler system and exists on your machine.



7. *make project* or *rebuild all*



8. Make sure to have the files *Z88.DYN*, *Z88COM.CFG, Z88O.OGL* and *Z88P.COL* in the same directory were you did the compilation i.e. where your executables are located. Otherwise, you don't need to wonder about fancy error massages. And the proper input files should exist here, too.

Name	Windows Sources	Windows- und UNIX Sources	Defines	Library
z88f	z88f.c ale88f.c easyfont.c tob88f.c wrim88f.c z88f.rc	apla88.c choy88.c cshe88.c dyn88f.c hexa88.c hpla88.c isod88.c lan88f.c lqua88.c m1.c m2.c qshe88.c ri188.c rw2y88f.c spla88.c spur88.c tetr88.c wlog88f.c wria88f.c z88a.c z88b.c z88cc.c z88f.h	FR_WIN95 FR_XDOUB FR_XLONG	comctl32.lib
z88i1	z88i1.c ale88i.c easyfont.c tob88i1.c wrim88i.c z88i1.rc	dyn88i1.c lan88i1.c ri188i.c w4y88i.c wlog88i1.c wria88i.c z88ai.c z88i.h	FR_WIN95 FR_XDOUB FR_XLONG	comctl32.lib
z88i2	z88i2.c ale88i.c easyfont.c tob88i2.c wrim88i.c z88i2.rc	apla88.c cshe88.c dyn88i2.c hexa88.c hpla88.c isod88.c jaco88.c lan88i2.c lqua88.c m1.c m2.c qshe88.c r1y88i.c r4y88i.c spla88.c spur88.c tetr88.c wlog88i2.c z88bi.c z88ci.c z88i.h	FR_WIN95 FR_XDOUB FR_XLONG	comctl32.lib
z88d	z88d.c ale88d.c easyfont.c tob88d.c wrim88d.c z88d.rc	dyn88d.c fuvs88.c lan88d.c m3.c m4.c riy88d.c sapl88.c scsh88.c shex88.c shpl88.c siso88.c slqu88.c span88.c sqsh88.c sspl88.c sspu88.c stet88.c wlog88d.c z88d.h	FR_WIN95	comctl32.lib
z88e	z88e.c ale88e.c easyfont.c tob88e.c wrim88e.c z88e.rc	apla88.c cshe88.c dyn88e.c forc88.c hexa88.c hpla88.c isod88.c lan88e.c lqua88.c m1.c m2.c qshe88.c riy88.c spla88.c spur88.c tetr88.c wlog88e.c z88e.h	FR_WIN95 FR_XDOUB FR_XLONG	comctl32.lib
z88n	z88n.c ale88n.c easyfont.c tob88n.c wrim88n.c z88n.rc	dyn88n.c join88.c lan88n.c mc188n.c mc288n.c mc388n.c mc488n.c mc588n.c rni88.c subn88.c wlog88n.c z88n.h	FR_WIN95	comctl32.lib
z88v	z88v.c ale88v.c g1i188.c g1i388.c g1i488.c g2i188.c g3i188.c g4i188.c g5i188.c gxi288.c tob88v.c z88v.rc	dyn88v.c lan88v.c wlog88v.c z88v.h	FR_WIN95	comctl32.lib
z88x	z88x.c ale88x.c easyfont.c tob88x.c wrim88x.c z88x.rc	dyn88x.c koi88x.c lan88x.c rea88x.c sub88x.c wlog88x.c wria88x.c z88fx.c z88tx.c z88x.h	FR_WIN95	comctl32.lib
z88g	z88g.c ale88g.c easyfont.c tob88g.c wrim88g.c z88g.rc	cosm88.c nast88.c lan88g.c wlog88g.c z88g.h	FR_WIN95 FR_XDOUB FR_XLONG	comctl32.lib
z88h	z88h.c ale88h.c easyfont.c tob88h.c wrim88h.c z88h.rc	lan88h.c rdy88h.c wlog88h.c z88h.h	FR_WIN95	comctl32.lib
z88p	z88p.c ale88p.c easyfont.c m8.c rcol88.c tob88p.c vc88.c vgpc88.c vgpp88.c vp88.c wlog88p.c z88p.rc	dyn88p.c lan88p.c m5.c m6.c m7.c pc88.c pgpc88.c pgpp88.c pp88.c z88p.h	FR_WIN95	comctl32.lib
z88o	z88o.c ale88o.c m11.c m13.c oglfont.c rogl88.c tob88o.c wlog88o.c z88o.rc	dyn88o.c lan88o.c m9.c m10.c m12.c oc88.c z88o.h	FR_WIN95	comctl32.lib opengl32.lib
z88com	z88com.c ale88c.c easyfont.c tob88c.c z88com.rc	lan88c.c wlog88c.c z88c.h	FR_WIN95	comctl32.lib

The Z88 executables and the matching sources, defines and libraries.

1.3 HOW TO INSTALL Z88 FOR UNIX AND LINUX

1.3.1 LINUX installation for RedHat (from 8.0) and SuSE (from 8.1)

Z88 for LINUX installs easily with RPM – the RedHat Package Manager- which is part of all well- known LINUX distributions. Login as *root* and proceed as follows:

Check if these packages are installed: Mozilla, Nedit and OpenMotif. How to check? Do this:

- *rpm –q mozilla*
- *rpm –q nedit*
- *rpm –q openmotif*

Install these programs if necessary. They are part of RedHat and SuSE LINUX.

Install Z88 by running RPM:

- *rpm –i z88-11.0-1.i386.rpm*

Now login as a normal user, change to any (working-) directory and launch Z88 by the command from an X-Terminal (i.e. a command window):

- *z88*

The Z88 commander Z88COM is started and important parameter files are loaded. When running *z88* for the first time, the first example is loaded, too. Therefore, you may instantly do your first Z88 calculation. The other examples reside in */usr/share/z88*. Put Z88COM and the X-term, which started Z88COM by the *z88* command, side-by-side or over-and-under to see both.

1.3.2 Installation for UNIX machines and other LINUX versions

If you've got an older or a newer LINUX system (but be sure to check if the RPM procedure (see above) works) or a true UNIX system, you are to compile Z88 at first. This is fairly easy as you will see below.

LINUX or UNIX installation in 4 steps:
For reasons of clarity all uppercased modules and file names are actually written in lower case, as is usual with UNIX.

1st step: Copy the Z88 files into a new or existing directory:
Simply put all Z88 files into an existing or new directory. Take care to do this as normal user and that you have read/write/execute permissions. This should be true for your home directory or an underlying subdirectory. Of course, it's all possible as superuser, too, but then paths must be adjusted. Again: Make sure that all permissions are properly set. Use *umask* if necessary. Internet distributions of Z88 feature only one single compressed file *z88.tar.gz*. Uncompress it:

- *gunzip z88.tar.gz*
- *tar -xvf z88.tar*

2nd step: Compile Z88 for UNIX or LINUX:
You need: C compiler, make, X11, Motif, OpenMotif or Lesstif

Any UNIX- C or C++ compiler should work properly. I found the GNU *gcc* and the compilers from SGI and HP doing a good job.

For LINUX: *COMPILE.LINUX* (mit Open Motif)
For SGI: *COMPILE.SGI* or for large SGI machines *COMPILE.ORIGIN*
For HP: *COMPILE.HP*
For others: Adjust one of the Make files *(*.mk.*)* and one of the *COMPILE.** files. Then adjust the file *Z88.FCD* so that the Motif programs Z88COM and Z88P are properly displayed.

For the experienced user (skip this for a first reading and proceed with the 3rd step)
This is the default procedure. On large computers you sometimes have the choice to use 8 Bytes instead of 4 Bytes for integers and 16 Bytes instead of 8 Bytes for floats. You may adjust this in the makefiles by *defines*:

Integer normal	Float normal	Integer extended	Float extended
FR_XLONG	*FR_XDOUB*	*FR_XLOLO*	*FR_XQUAD*
long	*double*	*long long*	*long double*
4 Bytes	*8 Bytes*	*8 Bytes*	*16 Bytes*
%ld	*%lf*	*%lld*	*%LF, %LE, %LG*

This is possible for the solver modules Z88F, Z88I1 and Z88I2 along with their subroutines. The following combinations are possible: FR_XLONG and FR_XDOUB (default), FR_XLONG and FR_XQUAD, FR_XLOLO and FR_XDOUB plus FR_XLOLO and FR_XQUAD.
For the rest of the Z88 modules (Z88COM, Z88D, Z88E, Z88G, Z88H, Z88N, Z88O, Z88P, Z88V) this is not completly coded because it makes no sense to run the plot program Z88P or the DXF converter Z88X with extended precision. Anyway, some of the sources are prepared for extended precision. But caution: The solver modules use the same finite element functions like the nodal force processor Z88E. Therefore, *two* makefiles do exist and it's a good idea to run them one after another:

Computer/OS	for the solver modules	for the remaining modules
LINUX	*z88.mk.kernel.linux*	*z88.mk.other.linux*
HP	*z88.mk.kernel.hp*	*z88.mk.other.hp*
SGI (-n32)	*z88.mk.kernel.sgi*	*z88.mk.other.sgi*
ORIGIN (-64)	*z88.mk.kernel.origin*	*z88.mk.other.origin*
others	analogously	analogously

The default procedure i.e. using the same *defines* for both the makefiles is as follows:

LINUX	*cp z88.fcd.linux z88.fcd*
	make -f z88.mk.kernel.linux kernel
	make -f z88.mk.other.linux other clean ready
SGI (-n32)	*cp z88.fcd.sgi z88.fcd*
	make -f z88.mk.kernel.sgi kernel
	make -f z88.mk.other.sgi other clean ready
...	... analogously

But if you want to use extended precision (FR_XLONG and FR_XQUAD or FR_XLOLO and FR_XQUAD) for the solver modules, but normal precision for the other modules

(FR_XLONG and FR_XDOUB, the default) then you must re- compile the modules used by Z88E (which were for the solvers compiled with extended precision) with normal precision or will confuse the linker:

LINUX	*cp z88.fcd.linux z88.fcd*
	make -f z88.mk.kernel.linux clean kernel
	make -f z88.mk.other.linux caution other clean ready
SGI (-n32)	*cp z88.fcd.sgi z88.fcd*
	make -f z88.mk.kernel.sgi clean kernel
	make -f z88.mk.other.sgi caution other clean ready
...	... analogously

3rd step: Enter your favourite Internet-Browser into Z88:
You should have installed a fancy browser on your system in order to display the Z88 online help. Use any internet browser e.g. *Netscape, Mozilla, Arena, Mosaic* or *Chimera:* Edit the header file *Z88.FCD*. Be sure to enter the proper browser prefix (keyword CPREFIX) matching your browser. The prefix tells the browser to load a specific HTML file from your machine rather from the Internet. For example:

- *Arena* does not need any prefix at all.
- *Mozilla*: *file:///home/yourname/z88/*, assuming that the Z88- HTML, the GIF- and the JPG-files are located in the directory */home/yourname/z88*

You can easily find out the prefix for your browser if you start it from an X-term with a Z88-HTML-file, e.g. *arena e88ix.htm* or *mozilla file:///home/yourname/z88/e88ix.htm*

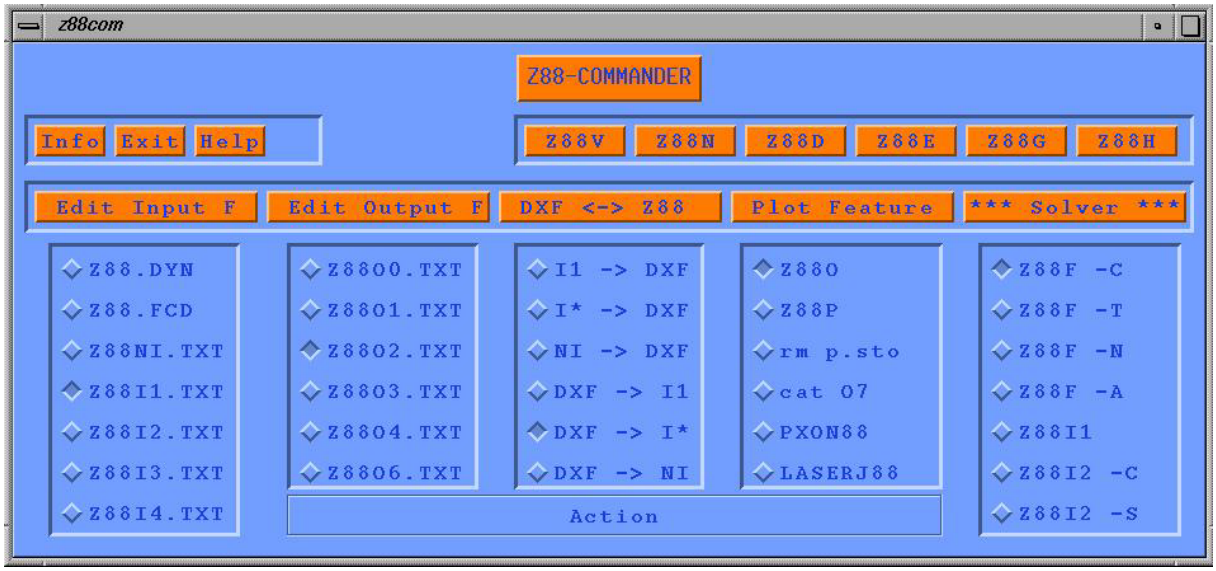
The help system is easy to use:

- Clicking the large *Z88 Commander* button invokes the directory for all Z88 chapters. Now enjoy browsing. If nothing happens - please wait a moment (some of those fat browsers need an endless time to load)
- Clicking the *Help* button invokes context sensive online-help: The Help button reverses its color indicating that help mode is active. Now click a command button to open the browser with the proper help chapter. Help mode stays active until you click the *Help* button again.

4th step: Enter your favourite editor into Z88:
You may use any ASCII editor. I found *joe* (WordStar-like) under LINUX a nice substitute for good old *vi*. *Nedit* is quite nice, too. Edit *Z88.FCD*.

And now: Run Z88:
You can start the various Z88 modules from a text console, from an X-term or by a shell-script. The Z88-Commander Z88COM and the plot programs Z88O and Z88P must be started on an X-Window surface like *Gnome, fvwm2, icewm, cde, kde*. Thus, it is good practice to launch all Z88 modules from an X-term using the Z88-Commander Z88COM ... so

Start your X-Window system, open an X-term and lauch Z88COM. Put Z88COM and the X-term, which started Z88COM, side-by-side or over-and-under to see both. The X-term is used for console input/output for the text-mode programs Z88F, Z88I1, Z88I2, Z88N, Z88D, Z88E, Z88X, Z88G, Z88H, Z88V.



If you are not pleased with my choice of colors and fonts, then edit the header file *Z88.FCD*. Be sure to store the original *Z88.FCD* file in order to have a ready-to-run file if something goes wrong as Z88COM and Z88P cannot run without a correct *Z88.FCD*.

... And how do you remove Z88 ?

If you've installed Z88 by RPM, simply type with *root* permissions: *rpm –e z88*.
If you've compiled Z88: Simply delete all files in the directory containing Z88. Then delete the directory if necessary.

1.4 DYNAMIC MEMORY Z88

HEADER FILE Z88.DYN AND FILECHECKER Z88V

All Z88 modules allocate memory dynamically. Although Z88 is delivered with default values in Z88.DYN the user may and should modify the values for best operation of Z88. The file Z88.DYN is there to be modified .

The language is defined also in Z88.DYN. Enter into a line, best located between DYNAMIC START and NET START, the key word **ENGLISH** or **GERMAN**.

Z88.DYN starts with the key DYNAMIC START and ends with DYNAMIC END. There is a section for the net generator (NET START, NET END), a common section for all modules (COMMON START, COMMON END), a section for the plot program (PLOT START, PLOT END) and an additional section for the Cuthill McKee program (CUTKEE START, CUTKEE END). Blank lines or comments are optional, only the uppercased keywords are recognized. After the keyword follows an integer value, separated by at least one blank. The order of the keywords is optional.

You can check the memory needs defined in Z88.DYN for the memory critical modules Z88F, Z88I1, Z88I2, Z88X, Z88N and Z88P with the Filechecker Z88V. Add approx. 200 K to the results for each program module, which is negligible under Windows and under UNIX.

A proper modification of Z88.DYN is definitely a good idea.

However, do not request unnecessarily much memory since this causes speed losses , especially when using virtual memory.

Test the memory needs for large structures. Proceed as follows depending on the solver:

The direct Cholesky solver Z88F:

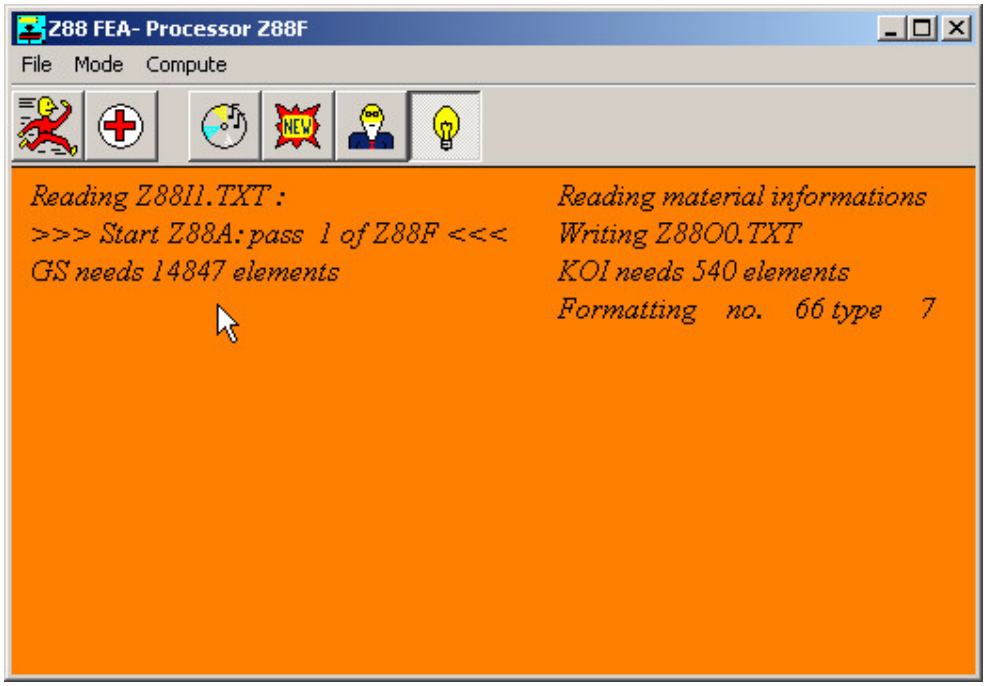
Windows NT/95: Z88F > Mode > Test Mode , Compute > Go
UNIX: z88f -t (console) or Z88F with option Test (Z88COM)

If you get here e.g. GS= 100,000, then enter, let's say, 120,000 for MAXGS in Z88.DYN but not 1,000,000 ! Then you estimate the total memory needs as described below or use Z88V.

Thus proceed for large structures for Z88 in 2 steps:

1st: State MAXGS
Windows NT/95: Z88F > Mode > Test Mode , Compute > Go
UNIX: z88f -t (console) or Z88F with option Test (Z88COM)

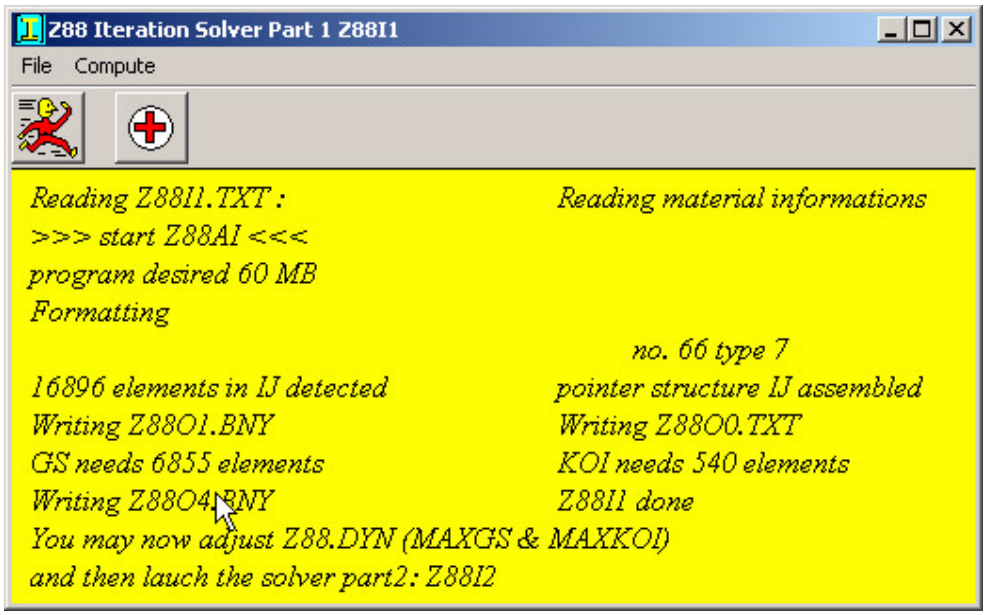
2nd: Correct Z88.DYN if necessary, state memory needs of Z88F with Z88V



(See the necessary memory MAXGS and MAXKOI, Windows. Looks similar on UNIX systems)

The iteration solver Z88I1 (part 1) and Z88I2 (part 2)

There's no test mode available because the first part of the iteration solver Z88I1 detects the memory needs for the second part Z88I2:



(See the necessary memory MAXGS and MAXKOI, Windows. Looks similar on UNIX systems)

However, the procedure for the iterations solver is quite tricky because you must define memory MAXSOR for the assembly of the sparse matrix. There is no way to pre- determine the needed memory but Z88I1 tells you if MAXSOR was too small. Then, increase MAXSOR in Z88.DYN and run Z88I1 again. Adjust MAXPUF to about 1/4 to 1/10 of MAXSOR. For example:

```
MAXSOR   5000000
MAXPUF    500000
```

Thus proceed for large structures for Z88 in 3 or more steps:

1st: run Z88I1

2nd: if Z88I1 completed properly, read off the values for MAXGS and MAXKOI and adjust Z88.DYN, if necessary. Now memory is proper adjusted for Z88I2.

3rd: if Z88I1 stopped because of lack of MAXSOR increase MAXSOR in Z88.DYN and run Z88I1 again. Adjust MAXPUF to about 1/4 to 1/10 of MAXSOR. Repeat this step until Z88I1 completes properly.

Make sure that your swap space is sufficient. Adjust if necessary:

Windows:
Start > Settings > Control Panel > System > Performance > Virtual Memory > Change. You choose the size of the permanent swap file depending on your own ideas.

UNIX:
Depending on the various UNIX operating systems the swap partition can be easily extended dynamically or an additional swap file must be created or the swap area must be deleted and a new swap area created with extended size.

All operating systems:

i recommend a size between 100 and 1000 MByte. The exact value depends on the actual size of your structures. If you get an error message "not enough dynamic memory" from a Z88 computing unit (mostly from Z88F), then increase the swap space.

There are no limits for the size of the structures for Z88. The maximum size is limited only by virtual memory of your computer and your imagination !

The Z88 modules check whether the predefined memory is sufficient for the current problem or if limits are reached and stop if necessary.

At commentless breakdown of a Z88 module check the accompanying .LOG file. Often the value for MAXKOI was too small !

Respect UNIX: If Z88 modules refuse to start, check the permissions of the .LOG files.

The delivered FEA example files work perfectly well with the delivered standard values in Z88.DYN. You will modify Z88.DYN if you want to calculate your own, large structures.

The .LOG files record the memory needs. Some more memory is needed for the program, local arrays and stack which one can neglect for Windows or UNIX.

Z88 normally deals with

Floating point numbers with Doubles = 8 bytes
Integers with Longs= 4 bytes.

However, on some UNIX machines you may compile (compiler directives FR_XQUAD and FR_XLOLO) the solver modules using
Floating point numbers with long Doubles = 16 bytes
Integers with long Longs = 8 bytes.
For specialists only!

Critical for the memory are Z88F, Z88I1, Z88I2, Z88P, Z88X and Z88N. To the modules Z88D, Z88E and Z88V applies: If Z88F or Z88I1 and Z88I2 run, then these modules will run, too. If in doubt, free your machine's memory by closing other unused programs.

The general description follows for Z88.DYN.

DYNAMIC start

Adjusting Language:
ENGLISH or **GERMAN**. If nothing is entered or the entry is wrong, English language is used automatically.

Section Net Generator:
NET START
MAXSE Maximum number of internal nodes for FE net generation. Must be clearly higher than produced FE nodes.
MAXESS Maximum number of super elements
MAXKSS Maximum number of super nodes
MAXAN Maximum number of nodes which can meet a super element. The default of 15 has proven well even for complex space structures with Hexahedrons No.10. May be increased in case of doubt.
NET END

Common Data:
COMMON START
MAXGS Maximum number of entries in the the total stiffness matrix. Actual number GS is recorded by Z88F and Z88I1.
MAXKOI Maximum number of entries in the coincidence vector = number nodes per element * number of finite elements. Example: 200 finite elements No.10 = 20 nodes per element * 200 = 4000. At mixed structures take the element type with most nodes and multiply by the number of elements. Required number of NKOI is recorded by Z88F and Z88I1.
MAXK Maximum number of nodes in the structure.
MAXE Maximum number of elements in the structure.
MAXNFG Maximum number of degrees of freedom in the structure.
MAXNEG Maximum number of material info lines for the structure.
MAXSOR For the iteration solver part 1 i.e. Z88I1 only. Z88I1 uses a sorting vector with the size of MAXSOR. There is no way to pre- determine the needed memory but Z88I1 tells you if MAXSOR was too small. In this case you must increase MAXSOR and launch Z88I1 again.
MAXPUF For the iteration solver part 1 i.e. Z88I1 only. This value tells

Z88I1 when to start another sorting. Adjust MAXPUF to about 1/4 to 1/10 of MAXSOR.

COMMON END

For plot program Z88P:

PLOT START

MFACCOMMON The following values from COMMON multiply by this factor: MAXKOI, MAXE, MAXK. Standard factor is 2. Useful for checks of input files which were produced by the net generator.

MAXGP Maximum number of Gauss points for stress plots. Example: 200 finite elements No.10, integration order 3: 3 * 3 * 3 = 27 Gauss points per element resulting in 27 * 200 = 5400 Gauss points in summary.

PLOT END

For the Cuthill- McKee program:

CUTKEE START

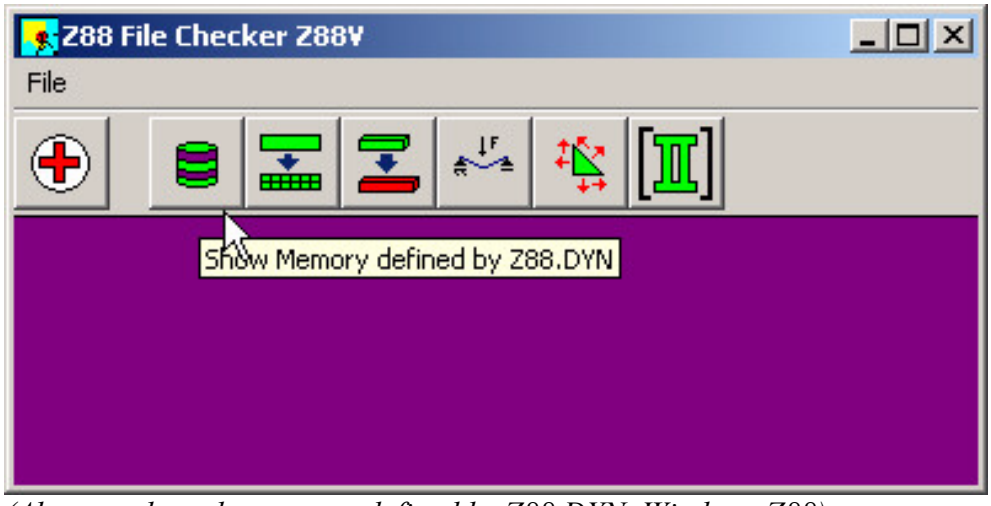
MAXGRA maximum degree of nodes

MAXNDL steps of the algorithm

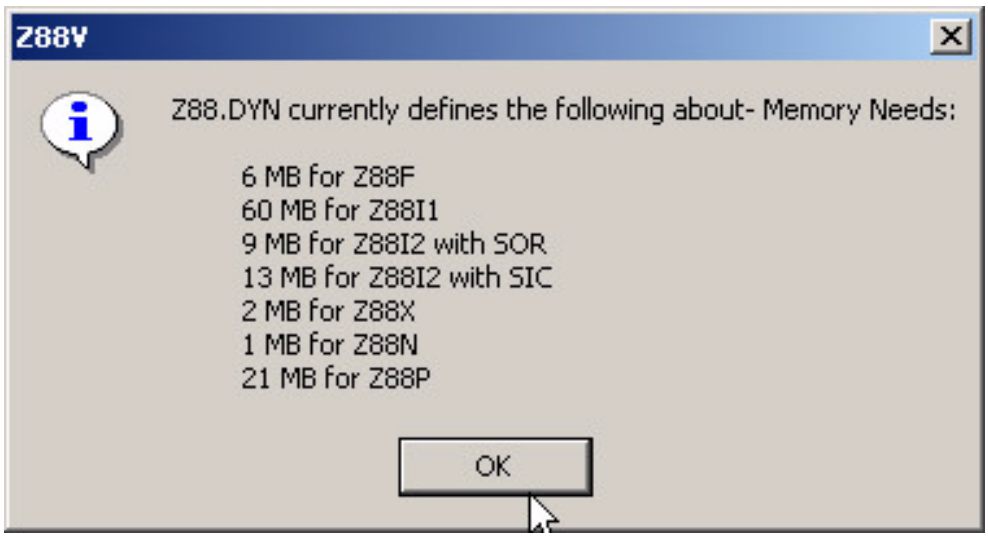
CUTKEE END

DYNAMIC END

You may state with Z88V which amount of memory the various Z88 modules will request.



(About to show the memory defined by Z88.DYN, Windows Z88)

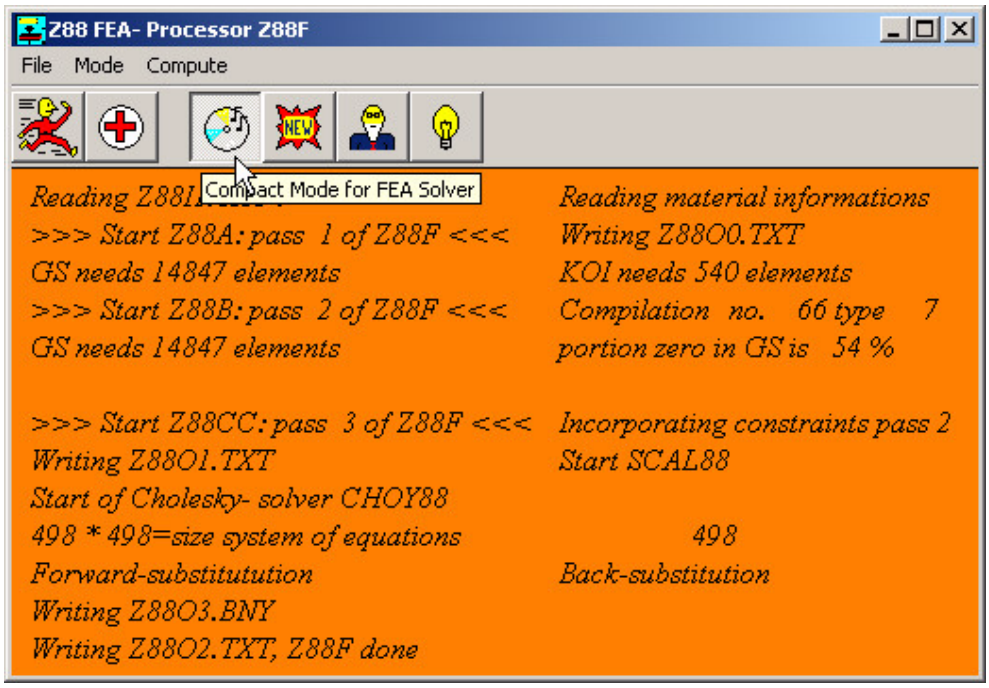

Z88V
Z88.DYN currently defines the following about- Memory Needs:
6 MB for Z88F
60 MB for Z88I1
9 MB for Z88I2 with SOR
13 MB for Z88I2 with SIC
2 MB for Z88X
1 MB for Z88N
21 MB for Z88P
OK

2 THE Z88 MODULES

2.1 THE DIRECT CHOLESKY SOLVER Z88F

NOTE: Always compare FEA calculations with analytical rough calculations, results of experiments, plausibility considerations and other tests without exeption !

The principal task of every FEA program is the calculation of the displacements. That's the job of Z88F. The calculated deflections are the starting point for a stress calculation with Z88D or nodal force calculation with Z88E.
Z88F is the right solver for small to medium structures. For large structures launch the iteration solver Z88I1 and Z88I2.
For the displacement calculation the FE processor Z88F can be started in different modes. This is chosen in Z88F in the Mode menu. *Compact Mode* is provided as default.



Note: The files Z88I1.TXT and Z88I2.TXT mentioned here are described more precisely in chapter 3.

(1) Compact Mode

Windows: Z88F > Mode > Compact Mode, Compute > Go
UNIX: z88f -c (console) or Z88F with option Compact (Z88COM)

Input files:
Z88I1.TXT (general structure data)
Z88I2.TXT (boundary conditions, constraints)

Output files:
Z88O0.TXT (processed structure data for documentation)

Z88O1.TXT (processed boundary conditions for documentation)
Z88O2.TXT (deflections)

In addition two binary files Z88O1.BNY and Z88O3.BNY are generated. These binary files are later used by Z88D (stress processor) and Z88E (nodal force processor).

A third binary file Z88O2.BNY can be generated in addition which contains the total stiffness matrix. By ist use the same structure can be computed with different boundary conditions without a new formatting and compiling process. This is provided by (2) *New Mode*. Because this file Z88O2.BNY can get very large, the generation of it can be suppressed with the option (1) *Compact Mode*. Prefer *Compact Mode* if

> Only one set of boundary conditions should be processed
> Computing of big structures (disk space !)

Basically, Compact Mode can always be used !

The other two modes *Z88F > Mode > New Mode* and *Z88F > Mode > Old Mode* for Windows XP/95 and *z88f -n* and *z88f -a* for the UNIX operating systzem can save computing time for a structure with several sets of constraints.

(2) New Mode

Windows: Z88F > Mode > New Mode, Compute > Go
UNIX: z88f -n (console) or Z88F with option New (Z88COM)

Input files:
Z88I1.TXT (general structure data)
Z88I2.TXT (boundary conditions)

Output files:
Z88O0.TXT (processed structure data for documentation)
Z88O1.TXT (processed boundary conditions for documentation)
Z88O2.TXT (deflections)

In addition two binary files Z88O1.BNY and Z88O3.BNY are generated. These binary files are later used by Z88D (stress processor) and Z88E (nodal force processor).

A third binary file Z88O2.BNY is generated in addition which contains the total stiffness matrix. By ist use the same structure can be computed with different boundary conditions without a new formatting and compiling process. The Old Mode *Z88F > Mode > Old Mode* reads in Z88O2.BNY, see below.

Prefer *Compact Mode* if

> Several sets of boundary conditions should be processed

Keep in mind that big structures can cause huge Z88O2.BNY files. Also see *Z88F > Mode > Old Mode*.

(3) Old Mode

Windows: Z88F > Mode > Old Mode, Compute > Go
UNIX: z88f -a (console, yes, lowercase A) or Z88F with option Old

Input files:
Z88I2.TXT (boundary conditions)
(Z88O2.BNY, was produced by a previous run *Z88F > Mode > New Mode* or *z88f -n*)

Output files:
Z88O1.TXT (processed boundary conditions for documentation)
Z88O2.TXT (deflections)
In addition the binary file Z88O3.BNY is filed. This binary file used by Z88D (stress processor) and Z88E (nodal force processor).

Z88F reads Z88O2.BNY containing the total stiffness matrix and the boundary conditions from Z88I2.TXT and solves the system of equations. The formatting and compiling process is skipped so that a given structure with different sets of constraints can be computed more quickly than with multiple runs of *Z88F > Mode > Compact Mode (z88f -c for UNIX)*. This Mode can be used only if *Z88F > Mode > New Mode (z88f -n for UNIX)* was run before.

Example:
General structure data Z88I1.TXT and

- 1st set of boundary conditions in Z88I2.TXT, *Z88F > Mode > New Mode*
- 2nd set of boundary conditions in Z88I2.TXT, *Z88F > Mode > Old Mode*
- (more sets of boundary conditions, *Z88F > Mode > Old Mode*)
- Last set of boundary conditions in Z88I2.TXT, *Z88F > Mode > Old Mode*

Modify the contents of Z88I2.TXT between the Z88F runs. Z88I2.TXT cannot maintain more than one set of constraints.

(4) Test Mode

Windows: Z88F > Mode > Test Mode, Compute > Go
UNIX: z88f -t (console) or Z88F with option Test (Z88COM)

Input files:
Z88I1.TXT (general structure data)

Output files:
Z88O0.TXT (processed structure data for documentation)

Only the file Z88O0.TXT (processed structure data for documentation) is produced along with the memory needs for total stiffness matrix and coincidence vector plotted on the screen. Use this mode for
> Checking the memory needs for MAXGS and MAXKOI.
> Checking if Z88F interprets Z88I1.TXT correctly and, as requested, puts the data in Z88O0.TXT.

2.2 THE ITERATION SOLVER Z88I1/Z88I2

NOTE:
Always compare FEA calculations with analytical rough calculations, results of experiments, plausibility considerations and other tests without exeption !

The principal task of every FEA program is the calculation of the displacements. That's the job of Z88I1 and Z88I2. The calculated deflections are the starting point for a stress calculation with Z88D or nodal force calculation with Z88E.

For large structures launch the iteration solver Z88I1 and Z88I2. Z88F is the right solver for small to medium structures.

The iteration solver uses only the so-called non- zero elements - this results in an absolute minimum for storage - and features two parts:

Iteration solver Part 1: Z88I1 builds the following pointers for the lower part of the total stiffness matrix GS:

- Pointer vektor IP points to the diagonal elements GS(i, i)
- Pointer vektor IEZ points to the column index GS(x, j)

A structure IJ will be assembled and then be sorted by a QSORT algorithm (an idea of Frank Koch). However, this step may need very much memory. Because of pure integer operations the computing speed is quite satisfying, though.

Example (ref. Schwarz, H.R: Methode der finiten Elemente) : Let the lower part of GS be:

GS(1,1)					
GS(2,1)	GS(2,2)				
	GS(3,2)	GS(3,3)			
GS(4,1)			GS(4,4)		
GS(5,1)		GS(5,3)		GS(5,5)	
	GS(6,2)		GS(6,4)		GS(6,6)

GS results in the following vector of non- zero elements:

GS(1,1)	GS(2,1)	GS(2,2)	GS(3,2)	GS(3,3)	GS(4,1)	GS(4,4)
GS(5,1)	GS(5,3)	GS(5,5)	GS(6,2)	GS(6,4)	GS(6,6)	

IEZ will result in:

1	1	2	2	3	1	4	1	3	5	2	4	6

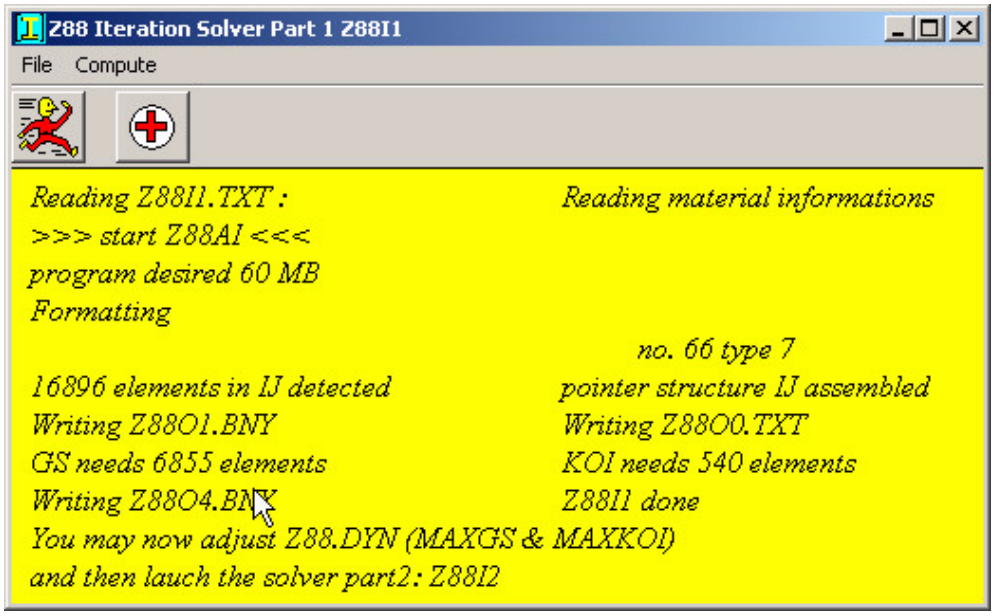
and IP:

1	3	5	7	10	13

The structure IJ holds MAXSOR elemente, ref. Memory definition file Z88.DYN. You must allocate memory MAXSOR for the assembly of the sparse matrix. There is no way to pre-determine the needed memory but Z88I1 tells you if MAXSOR was too small. Then, increase MAXSOR in Z88.DYN and run Z88I1 again. Adjust MAXPUF (for intermediate sorting) to about 1/4 to 1/10 of MAXSOR. For example:

```
MAXSOR  5000000
MAXPUF   500000
```

Z88I1 stores both the pointer vectors in a binary file Z88O4.BNY, which may become quite large.



Z88I1 tells you how much memory for GS (= MAXGS) and for KOI (= MAXKOI) you must allocate; adjust this in Z88.DYN. See an example of Z88.DYN:

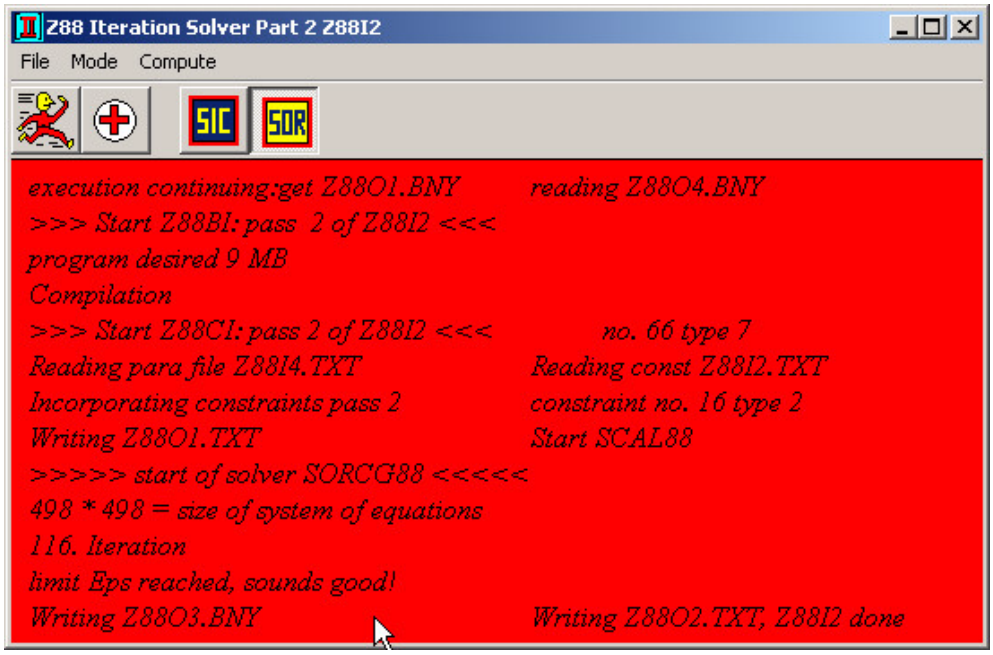
```
COMMON START
    MAXGS     600000   ← adjust this before running Z88I2
    MAXKOI    132000   ← adjust this before running Z88I2
    MAXK       11000
    MAXE       33000
    MAXNFG     32000
    MAXNEG        32
    MAXSOR   5000000   ← adjust this before running Z88I1
    MAXPUF    500000   ← adjust this before running Z88I1
  COMMON END
```

Thus proceed for large structures for Z88 in 3 or more steps:
1st: run Z88I1

2nd: if Z88I1 completed properly, read off the values for MAXGS and MAXKOI and adjust Z88.DYN, if necessary. Now memory is proper adjusted for Z88I2.

3rd: if Z88I1 stopped because of lack of MAXSOR increase MAXSOR in Z88.DYN and run Z88I1 again. Adjust MAXPUF to about 1/4 to 1/10 of MAXSOR. Repeat this step until Z88I1 completes properly.

Iteration solver Part 2: Z88I2 computes the element stiffnes matrices, compiles the total stiffness matrix, incorporates the boundary conditions, scales the system of equations and solves the (huge) system of equations by the conjugate gradient algorithm. Preconditioning is done for better convergence. Choose your favorite preconditioner: Either a SOR step or a so-called incomplete Cholesky decomposition. Default is *SOR* preconditioning, needs lesser memory, too. Choose incomplete Cholesky decomposition (*shiftet incomplete Cholesky decomposition SIC*) only in special cases.



(1) Conjugate Gradients with SOR preconditioning

Windows: Z88I2 > Mode > Precon: Overrelaxation, Compute > Go
UNIX: z88i2 -s (console) or Solver: Z88I2 -S (Z88COM)

(2) Conjugate Gradients with SIC preconditioning

Windows: Z88I2 > Mode > Precon: Inco. Cholesky decom., Compute > Go
UNIX: z88i2 -c (console) or Solver: Z88I2 -C (Z88COM)

In addition you must supply three entries in the parameter file Z88I4.TXT:

- termination criterion: maximum count of iterations(for example 10000) reached
- termination criterion: residual vector < limit *Epsilon* (for example 1e-7)
- parameter for the convergence acceleration. Meaning depends on the preconditioner:

1. in case of SOR: Relaxation factor *Omega* (from 0 to 2, good values may vary from 0.8 to 1.2).
2. in case of SIC: Shift factor *Alpha* (from 0 to 1, good values may vary from 0.0001 to 0.1). For further information consult the special literature)

Note: The files Z88I1.TXT, Z88I2 and Z88I4.TXT mentioned here are described more precisely in chapter 3.

Input files:

Z88I1.TXT (general structure data)
Z88I2.TXT (boundary conditions, constraints)
Z88I4.TXT (parameter file for the iteration solver part 2: Z88I2)

Output files:

Z88O0.TXT (processed structure data for documentation)
Z88O1.TXT (processed boundary conditions for documentation)
Z88O2.TXT (deflections)

In addition two binary files Z88O1.BNY and Z88O3.BNY are generated. These binary files are later used by Z88D (stress processor) and Z88E (nodal force processor).

2.3 THE STRESS PROCESSOR Z88D

A stress calculation with Z88D can run only if the deflections were calculated by Z88F or Z88I1 and Z88I2 before. The stress calculation is independent of the nodal force calculation.

Z88D is controlled by the parameter file Z88I3.TXT .

It fixes:

> Calculation of the stresses at the Gauss points or at the corner nodes
> Additional calculation of radial and tangential stresses for elements No. 3, 7, 8, 11, 12, 14 and 15.
> Calculation of von Mises stresses for continuum elements No. 1, 3, 6, 7, 10, 11, 12, 14, 15, 16, 17, 18, 19 and 20.

Format of for Z88I3.TXT see chapter 3.

The results of Z88D are presented in the file Z88O3.TXT

2.4 THE NODAL FORCE PROCESSOR Z88E

A nodal force calculation with Z88E can run only if the deflections were calculated by Z88F or Z88I1 and Z88I2 before. Nodal force calculation is independent of the stress calculation.

The nodal forces are calculated separately for each element . If several elements meet a node, one gets the complete nodal force for this node by adding the nodal forces of all accessing elements. This results are presented in the file Z88O4.TXT, too.

The results of Z88E are presented in the file Z88O4.TXT.

2.5 THE MESH GENERATOR Z88N

The mesh generator Z88N can produce 2-dimensional and 3-dimensional meshes. Z88N reads the mesh generator input file Z88NI.TXT and writes the general structure data file Z88I1.TXT.

For the description of Z88NI.TXT see chapter 3.

A mesh generation is sensible and permitted only for continuum elements:

Super structure	Finite element structure
Plane stress element No.7	Plane stress element No.7
Torus No.8	Torus No.8
Plane stress element No.11	Plane stress element No.7
Torus No.12	Torus No.8
Hexahedron No.10	Hexahedron No.10
Hexahedron No.10	Hexahedron No.1
Plate No.20	Plate No.20
Plate No.20	Plate No.19

Mixed structures e.g. containing Plane Stress Elements No.7 and Trusses No.9, cannot be processed.

In such a case let the mesh generator process a super structure containing only Plane Stress Elements No.7 without any Trusses No.9. Run Z88N. Then convert with the CAD converter Z88X the file Z88I1.TXT generated by the mesh generator Z88N to the DXF file Z88X.DXF. Start your CAD programm, import Z88X.DXF and insert the trusses, in addition you can also define the constraints on the fly. Export the drawing to Z88X.DXF, run Z88X again and generate Z88I1.TXT (general structure data) and optionally Z88I2.TXT (boundary conditions).

Mode of operation of the mesh generator:
The generation of FE meshes proceeds as follows: The continuum is described by super elements (short SE), which practically corresponds to a quite rough FE structure. Super elements can be: Hexahedrons No.10, Plane Stress Element No.7 and Plane Stress Element No.11 as well as Toruses No.8 and Toruses No.12 and Plates No.20.

The super structure then will be subdivided. This is done super element wise, starting with SE 1, SE 2 up to the last SE. SE 1 produces the finite elements (short FE) 1 to j, SE 2 the FE j+1 to k, SE 3 the FE k+1 to m and so on. Within the SE the direction of the local coordinates determines the nodal numbers and the element numbers of the FE structure. Definition:

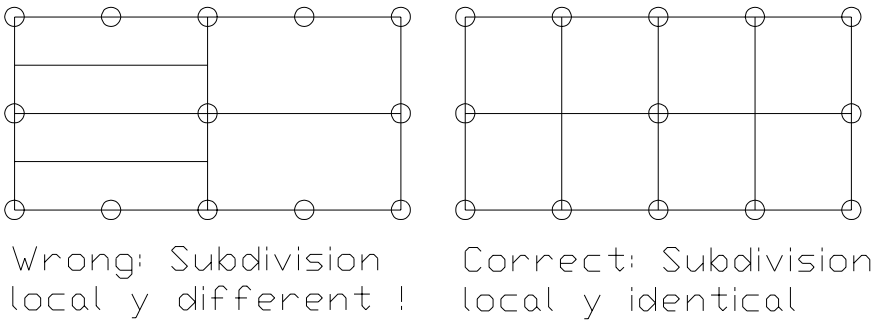
> Local x axis runs in direction of local nodes 1 and 2
> Local y axis runs in direction of local nodes 1 and 4
> Local z axis runs in direction of local nodes 1 and 5

Super structures in space are subdivided first in z, then in y and for the end in x direction i. e. the FE element numbers start along the z direction. To plane and axially symmetric structures applies analogously: The numbering starts along the y axis or for axially symmetric elements along the z axis (cylinder coordinates!).

Along the local axises can be subdivided as follows:

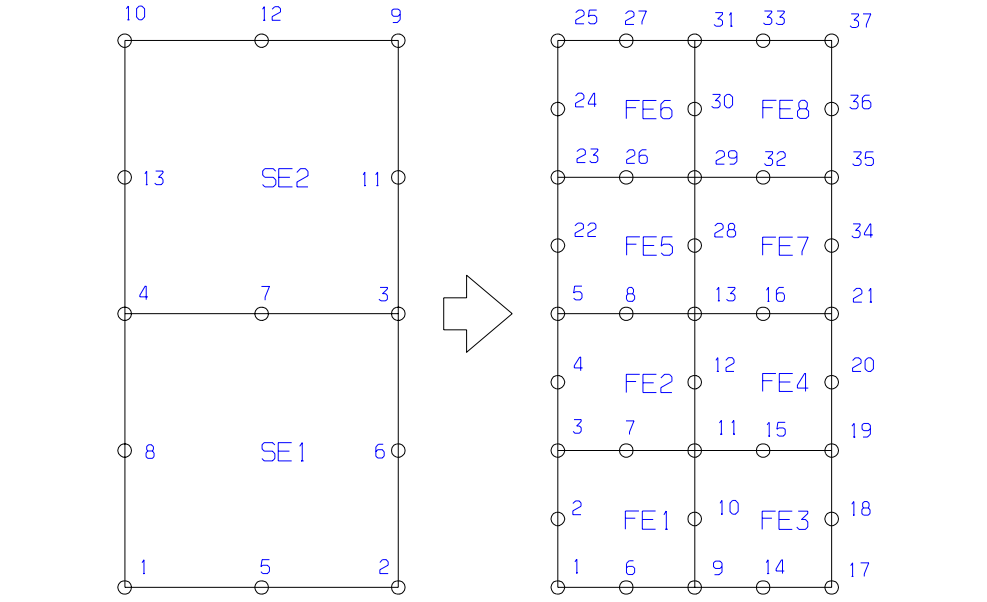
> Equidistant
> Increasing geometrically from node 1 to 4 or 5: Mesh becomes rougher
> Decreasing geometrically from node 1 to 4 or 5: Mesh becomes finer

It is obvious, that for lines or areas, which two super elements share, the super elements must be subdivided exactly the same! The mesh generator doesn't check this and then generates useless or totally mad FE meshes. Example:



Because the local axises x, y and z are defined by the location of the local nodes 1, 4 and 5, it is possible to generate almost arbitrary numberings for nodes and elements of the FE structure by corresponding construction of the coincidence list in the mesh generator input file Z88NI.TXT.

Example for the generation of a FE structure with 8 FE Plane Stress Elements No.7 from a super structure with 2 Plane Stress Elements No.7 (looks the same with Toruses No.8):



Coincidence of 1st superelement: 1-2-3-4-5-6-7-8
Coincidence of 2nd superelement: 4-3-9-10-7-11-12-13

Specials:

The mesh generator checks which nodes are already known at the production of new FE nodes. It needs for this check a trap radius (a computer cannot meet a floating point number exactly). This trap radius is provided for all 3 axises per default 0.01. Modify the trap radiuses when processing very small or very large numerical values.

In addition the mesh generator determines for each super element, which other super elements meet this super element. For Plane Stress Elements No.7 and No.11 or Toruses No.8 and No.12 this can be at the most 8 other SE. This maximum number MAXAN is provided in Z88.DYN per default 15. Theoretically, Hexahedrons No.10 can meet 26 other elements (6 areas, 8 corners, 12 edges). Practice has proved, that even complicated space structures with MAXAN = 15 worked fine. If in doubt increase MAXAN in Z88.DYN.

Attention mesh generator Z88N: The generator can generate input files with no trouble at all which blast all limits of the FE processor. Generate therefore at first rougher FE structures, check with *Z88F Test Mode* whether they fit into memory, then refine if necessary. A good starting point: Produce approx. 5..10 times more finite elements than super elements.

Note mesh generator Z88N: If coordinate flag KFLAG is set to 1 in the mesh generator input file Z88NI.TXT i.e. input values are polar or cylindrical coordinates, then the generated output file Z88I1.TXT (general structure data) has always cartesian coordinates and KFLAG is set to 0.

2.6 THE PLOT PROGRAM Z88P

The more modern Z88 plot program is the new OpenGL plot program Z88O, ref. Chp. 2.11. However, the approved plot program Z88P stays part of the Z88 FEA system.

The plot program Z88P plots undeflected, deformed, both undeflected and deflected structures as well as super structures.

Z88P can plot on the screen or into a file. The plot file contains so-called HP-GL commands which are used by HP plotters. Of course, it is optional to process the HP-GL files also in other programs, e.g. in CorelDraw, WinWord etc. If necessary change the ending .TXT. E.g. Winword expects the ending . HGL or .PLT for HP-GL files.

In addition, *von Mises* stresses can be plotted on the screen or send to a plotter. The screen colors can be adjusted for Windows in the file Z88P.COL. The file Z88.FCD for the UNIX operation system allows for much more possibilities in changing colors and fonts as well as the sizes and locations of the pushbuttons, radioboxes etc. You can completely modify the whole appearance of Z88P for UNIX as you wish.

Needed Files	Super structures	Undeflected FE struc.	Deflected FE struct.
Z88NI.TXT	Yes	No	No
Z88I1.TXT	No	Yes	Yes
Z88O2.TXT	No	No	Yes
Z88O5.TXT	No	Yes, for Mises stresses	No

For fastest operation Z88P connects the nodal points with straight lines, although for Serendipity elements 7, 8, 10, 11 and 12 the edges of the elements are square or cubic curves.

Note: You can also use your CAD program together with CAD converter Z88X to plot super structures and undeflected finite element structures, yet deflected structures and von Mises stresses do need Z88P. Or convert Z88's plot files (HP-GL language) into DXF files: CorelDraw does a very good job.

Z88P saves the last structure file name, the several factors and label adjustments in a file Z88P.STO. This file is loaded when starting Z88P, so that the last structure is restored automatically. If you like to start with a new structure, the file Z88P.STO should be deleted before launching Z88P. This can be done with *Plot > Delete Z88P.STO* (Windows) or *Plot* with option *rm z88p.sto* (UNIX) in the Z88-Commander Z88COM.

Special key strokes for Windows:

Prior:	increase zoom
Next:	decrease zoom
Cursor left:	panning X direction
Cursor right:	panning X direction
Cursor up:	panning Y direction
Cursor down:	panning Y direction

..And in addition for 3-D structures for Windows:

Home: panning Z direction
End: panning Z direction
F2: rotate around X axis
F3: rotate around X axis
F4: rotate around Y axis
F5: rotate around Y axis
F6: rotate around Z axis
F7: rotate around Z axis
F8: reset all rotations to 0

Under UNIX use the usual X and Motif key assignments: Tab key and arrow keys for choice and space bar for activation.

Pen plotters draw the undeflected structure with pen 1 and the deflected structure with pen 2.

Explanation of some menu items:

Name of Structure File:

Windows: ***File > Structure File***
UNIX: ***Stru.*** **text field directly on window**

Choose the structure file here. Enter name, if necessary with path, press return. The new stucture is loaded and drawn immediately.

Name of Plotter File:

Windows: ***File > Plotfile***
UNIX: ***Plot.*** **text field directly on window**

Choose a plotter file name here. Default name is Z88O6.TXT. The plot file includes HP-GL commands in ASCII code ready for reprocessing by other programs, e.g. CorelDraw.

Windows: Serial plotters: A direct output from Z88P to a plotter does not always work. If necessary, put Xon/Xoff sequence in Z88O6.TXT with the Z88-Commander after producing Z88O6.TXT. Then Z88O6.TXT can be send to a serial plotter by

copy /B Z88O6.TXT com1: (or com2:)

You also can use the Hyperterminal of Windows. Try ASCII, CR-LF ending, local echo, line delay 10 msec, byte delay 1 msec. However, large files can cause timeouts. This is a generally poor attribute of serial interfaces, cf. the notes below for UNIX.

Windows: HP LaserJet: Some laser printers can be switched from PCL mode to HP-GL mode by hand, for others this works unfortunately only by software. In this case put a LaserJet sequence in Z88O6.TXT with the Z88-Commander Z88COM after producing Z88O6.TXT. Then do normal printing.

UNIX: Serial plotters: Serial interfaces are adjusted with the stty-command as **root**

stty sane ixon ispeed 9600 cs8 -cstopb -parenb < /dev/ttyS1

Here /dev/ttyS1 is the second serial interface. The first serial interface is /dev/ttyS0 . If your plotter needs a software switchover on protocol Xon/Xoff, start the little utility *pxon88* after the production of Z88O6.TXT by Z88P . Then send Z88O6.TXT as **root** to the interface:

cat z88o6.txt > /dev/ttyS1

You can also put the serial plotter as a raw-device into /etc/printcap directly. Lets say it would be called HP7475A-a3-raw there. As a normal user you can then plot via the UNIX spooling system

lpr -PHP7475A-a3-raw z88o6.txt

Caution: Serial pen plotters are extremely slow devices, i. e. they can plot a lot due to the very efficient HP-GL language with extremely little information. For huge HP-GL files this can have the effect that the serial interface - despite Xon/Xoff - delivers far too *few* bytes to the plotter and therefore gets a timeout. See these general difficulties for LINUX under /usr/doc/howto at printer-howto and serial-howto.

UNIX: LaserJet: Some laser printers can be switched from PCL mode to HP-GL mode by hand, for others this works unfortunately only by software. In this case put a LaserJet sequence in Z88O6.TXT by the little utility *laserj88* after producing Z88O6.TXT with Z88P. Take care that laser printers have a *raw* entry in /etc/printcap as the UNIX spooling system is supposed to pass the HP-GL files completely unfiltered. Plot using

lpr -Praw z88o6.txt

Notes Windows and UNIX: The HP-GL commands produced by Z88 work faultlessly on several HP and IBM plotters if the physical adjustments are fixed correctly. Check out if your plotter is actually 100 % HP-compatible !

Another predefined file name is Z88O7.TXT. You can store the undeflected structure in Z88O6.TXT and the deflected structure in Z88O7.TXT if you do not want to do it by the Z88P command *Structure > Un- and Deflected*. The file Z88O7.TXT can be appended alternatively to Z88O6.TXT later allowing the plotting of the undeflected and the deflected structure in one run. Mind the order: Store Z88O6.TXT first, then Z88O7.TXT. But you can also plot completely different structures.

Deformation Modes of the Structure:

Windows: ***Structure > Undeflected, Deflected, Un- and Deflected***
UNIX: **Radio box** ***Undefle., Deflected, Un+Defle.***

Plots the undeflected structure, the deflected structure or both. Stresses can be plotted only on undeflected structures. When using *Un- and Deflected* nodes and element labels are plotted on the undeflected structure.

Caution Deflected and Un- and Deflected: The user must have executed a calculation of displacements before using this function. Do a FEA run with Z88F before using Z88P. Otherwise, some old files Z88O2.TXT (displacements) from earlier Z88 runs are opened

causing totally wrong results !!

Output in Plotter File:

Windows: ***Output > CRT, Plotter***
UNIX: **Pushbutton** ***Plot.***

Default is plotting on the screen (CRT). Selecting *Plotter* produces a HP-GL file whose name has been selected with *Interface* at choice. This runs very fast. Z88P shifts back again immediately to CRT after storing the HP-GL file and after quitting the message box.

Choice of View:

Windows: ***View > XY, XZ, YZ, 3-Dim***
UNIX: **Radio box** ***XY, XZ, YZ, 3D***

Select view depending on structure: At 2-dimensional structures XY, at 3-dimensional structures 3-Dim. Mind 3-D structures: The desired side views can be shown with XY, XZ and YZ, however, the nodes and element labels are plottet ascendingly (starting with node 1 or element 1 ascending to the last node or element), the stress points in the order of the Gauss points, ascending from element to element. The finished pictures can therefore show nodes and element numbers as well as stresses which does not really lie in the level of the side view! Only 3-D view gives reliable information. Note: If starting with a "fresh" 3D- structure in Z88P you will see under Windows the menu hook improperly set on XY or under UNIX the radio button XY set, because the menus are built up before reading the Z88 files. Do not care.

Drawing Node and Element Numbers:

Windows: ***Labels > No Labels, Elements, Nodes, Label All***
UNIX: **Radio box** ***No Labels, Elements, Nodes, Both***

Plot the element numbers or the node numbers or both or skip numbering. Labeling can get confusing because, depending on view, different numbers of complicated space structures are plotted several times at the same location. Carefully place structure parts by corresponding rotations.

Coordinate System:

Z88P uses a coordinate system which starts in the center of the CRT or the center of the sheet. It is fixed as follows:

Axis	CRT min	CRT max	Plotter min	Plotter max
X	-100	+100	-138	+138
Y	-100	+100	-100	+100
Z	-100	+100	-100	+100

For 3-D structures the transformation of the 3-D coordinates to the 2-D CRT or plotter is carried out via an isometric view ref. ISO 5. If the rotation angles ROTX, ROTY and ROTZ are set to 0 respectively, then is valid: X : Y : Z = 1 : 1 : 1. Z points up and X and Y are tilt by 30 degrees.
Windows: The rotation angles can be modified by steps of 10 degrees with the buttons *F2* to *F7* or with *Factors > Rotations 3-D* in optional values. *F8* resets all rotation angles to 0

degrees.

UNIX: The rotation angles can be modified by steps of 10 degrees with the Pushbuttons *RX+, RX-, RY+, RY-, RZ+* and *RZ-*. Pushbutton *Rot 0* resets all rotation angles to 0 degrees.

Plots fitting entirely on the screen usually work with the same factors for the plotter. However, because plotters have different X-Y ratios, factors need to be changed slightly for plotter output sometimes.

Global Magnifications:

Windows: ***Factors > Global Magnifications***

Do zooming either in steps with the keys *Prior* and *Next* or delicately with factors FACX, FACY, FACZ. The entry of factors is also useful if several structures should be plotted using the same factors.

UNIX: **Pushbuttons *Zoom+* and *Zoom-***

Panning:

Windows: ***Factors > Centre Factors***

Do panning into X direction with *left* and *right cursor keys*, into Y with *cursor up* and *cursor down* and into Z (at three-dimensional structures) with *Home* and *End*. Alternatively you can use *Factors > Centre Factors* : CX, CY and CZ.

UNIX: **Pushbuttons *X+, X-, Y+, Y-, Z+, Z-***

Enlarging Deflections:

Windows: ***Factors > Deflections***
UNIX: **Text fields *FUX, FUY* and *FUZ***

Enlarge the computed deflections with the factors FUX, FUY and FUZ. Default values are 100. Attention UNIX: As usual for UNIX, the changes only apply at a respective *Return*. However, you can type in all three fields without *Return* and then press the Pushbutton *Regen* (for Regenerate).

Rotations:

Windows: ***Factors > Rotations 3-D***

The rotations around X, Y and Z axises are defined with Factors > Rotations 3-D : ROTX, ROTY and ROTZ. Default values are 0. With the F2 . . F7 keys the structure can be revolved in steps of 10 degrees.

UNIX: **Pushbuttons *RX+, RX-, RY+, RY-, RZ+, RZ-***

Turn in steps of 10 degrees. Pushbutton *Rot 0* resets the rotations to 0.

Height Ratio:

Windows: ***Factors > X-Correction FXCOR***
UNIX: **Text field *FXCOR***

With the function X-Correction FXCOR the height ratio can be adjusted to the monitor customization. The default values are 0.75 for NT and 0.85 for UNIX. Modify FXCOR depending on monitor type if necessary. FXCOR is stored in Z88P.STO for further use.

Von Mises Stresses:

Windows: ***Mises Stresses > No von Mises stresses, Show von Mises stresses***
UNIX: **Pushbutton *Mises***

If you have calculated von Mises stresses by Z88D (only useful and possible for continuum elements of the type 1, 3, 6, 7, 8, 10, 11 or 12) then you can plot the von Mises stresses either into the Gauss points (elements No.1, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19 and 20) or into the center of gravity of the elements (elements No.3 and 6). Possible for CRT and Plotter.

Stresses will only show if you have chosen *Structure* > *Undeflected*. Z88P otherwise recalls that you wanted to plot von Mises stresses and shows them when switching from *Show von Mises stresses* to *Structure* > *Undeflected.*

If you choose *Show von Mises stresses* and then *Output* > *Plotter* the von Mises stresses are converted into a letter scale from A to J to match even ancient pen plotters with only 2 or 6 pens or monochrome laser printers.

On the screen the stresses are plottet using a color scale. You can adjust this color range in Z88P.COL (Windows) or Z88.FCD (UNIX) to match your personal wishes.

Consult the output file Z88O3.TXT for exact stress results because Z88P shows stresses only in a 10-step range.

Caution: The operator is responsible for first running a stress calculation by Z88D before using this function. Run Z88F or the Iteration solver for a calculation of the displacements and then Z88D for calculating the stresses before starting Z88P. Otherwise some old stress files Z88O5.TXT from earlier calculations are read causing totally wrong results !!

Keep in mind:

- Calculate stresses with Z88D. Enter 1 for the von Mises flag ISFLAG in the file Z88I3.TXT and a value greater than 0 for the integration order INTORD in Z88I3.TXT when using elements 1, 7, 8, 10, 11 and 12.
- Von Mises stresses are only computed for continuum elements of type 1, 3, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19 and 20.
- Switch to *Undeflected*.
- Take into account: For showing a "nice" stress scale make the stresses in the range from 0 to 100 or from 0 to 1,000. If necessary enter modified boundary conditions and recalculate to real results. This is because the stress scale is built of integer results.

Automatic Scaling:

Windows: ***Autoscale > No Autoscale, Yes Autoscale***
UNIX: **Pushbutton *AutoS***

The Autoscale function takes care that structures will completely fit on the screen.

Autoscale activates automatically if no file Z88P.STO is available or if a new structure is loaded by *File > Structure File*. Autoscale again is deactivated immediately and the hook points to *No Autoscale*. If a file Z88P.STO is present the factors are read from this file. You can then scale properly with *Autoscale > Yes Autoscale*. However, Autoscale switches again immediately to *No Autoscale*. *Autoscale > Yes Autoscale* is a kind of pushbutton. The explanation above corresponds to UNIX as well.

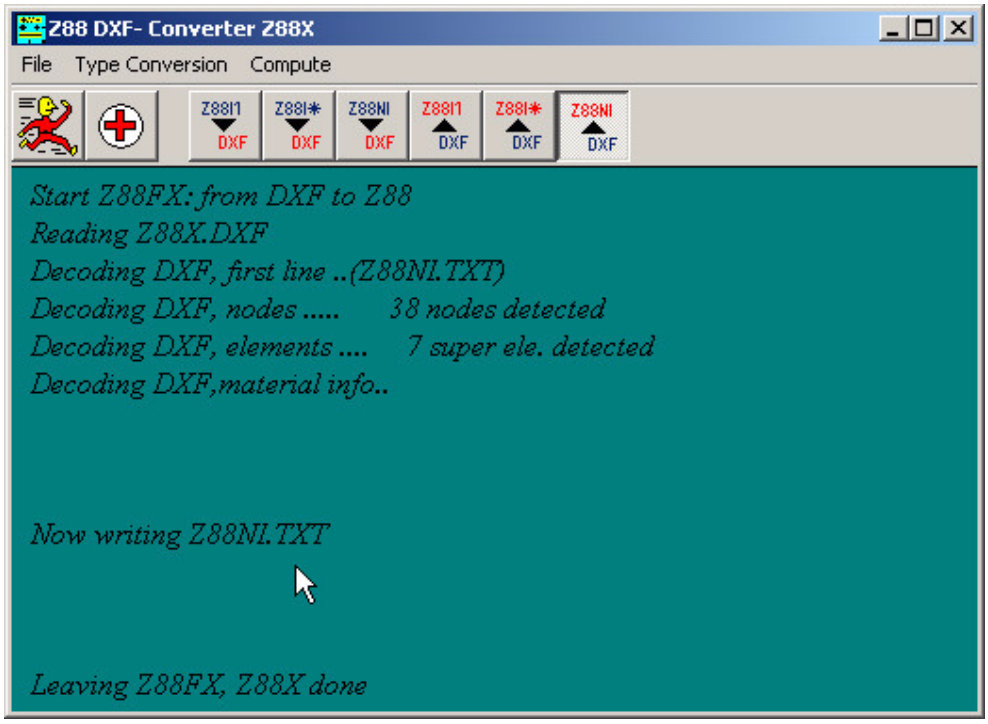
2.7 THE CAD CONVERTER Z88X

2.7.1 OVERVIEW Z88X

The CAD converter Z88X works in two directions:

(I) You design your component in a CAD system and generate Z88 data. You cover in the CAD system your component with a FE mesh or a super-structure following certain rules which follow below, and add if necessary boundary conditions and material informations. Then make your CAD system generating a DXF file and start the CAD converter Z88X. The Z88 entry files are produced by Z88X and you can start with the FE analysis.

Windows:
Z88X, > Type Conversion > 4 from Z88X.DXF to Z88I1.TXT
Z88X, > Type Conversion > 5 from Z88X.DXF to Z88I . TXT (default)*
Z88X, > Type Conversion > 6 from Z88X.DXF to Z88NI.TXT
... and > Compute > Go



UNIX:
z88x -i1fx (Z88X.DXF to Z88I1.TXT, "I1 from X")
z88x -iafx (Z88X.DXF to Z88I . TXT, "I all from X",)*
z88x -nifx (Z88X.DXF to Z88NI.TXT, "NI from X")
... or use the Z88-Commander with the proper option for Z88X

(II) Convert your Z88 entry files into CAD data. This is very interesting for Z88 data sets already existing, for controls, for completions of the FE structure, but also for plotting the FE structure by CAD program.

Windows:
Z88X, > Type Conversion > 1 from Z88I1.TXT to Z88X.DXF
Z88X, > Type Conversion > 2 from Z88I . TXT to Z88X.DXF*
Z88X, > Type Conversion > 3 from Z88NI.TXT to Z88X.DXF
... and > Compute > Go

UNIX:
z88x -i1tx (Z88I1.TXT to Z88X.DXF, "I1 to X")
z88x -iatx (Z88I . TXT to Z88X.DXF, "I all to X",)*
z88x -nitx (Z88NI.TXT to Z88X.DXF, "NI to X")
... or use the Z88-Commander with the proper option for Z88X

Since the converter is completely compatible in both directions, you can execute the possibilities I and II in succession as you wish. You will not find any data loss!

That makes a most interesting variant:

(III) Mixed Operation, e.g.

- Component-and super-structural layout done in CAD program
- Conversion CAD ---> Z88
- Meshing in Z88
- Conversion Z88 ---> CAD
- Complete FE structure in CAD e.g. with not-mesh generator capable elements
- Conversion CAD ---> Z88
- Change e.g. material informations in Z88
- Conversion Z88 --- > CAD
- Installation of the boundary conditions in CAD
- Conversion CAD ---> Z88
- FE analysis in Z88
- Etcetera

Which CAD systems can cooperate with Z88 ?

Well, any CAD systems which can import (read) and export (write) DXF files. However, we cannot guarantee any success as some of the CAD guys are changing their DXF definitions from month to month. Z88 V10 has been intensively tested together with the different AutoCAD LT versions for Windows of Autodesk, and AutoDesk's DXF guidelines have been rgarded as the inventor of the DXF interface, according to AC1009 and AC1012.

The general philosophy of a CAD - FEA data interchange:

CAD files containe nondirectional informations. It is only a wild collections of lines, points and texts, stored in the order of its production to make things worse.
Basically, a FEA system needs topological information which most CAD systems cannot supply. The FEA system must know that these and those lines form a finite element and that these and those points are included in this element. This could be made on principle if one would design in the CAD system in a quite firmly predefined order. Experiments showed that, indeed, this is possible for very simple components, but it will not work for complex components. And, yes, this is what one wants to do in practice: FE analysis on complex

structures !

These difficulties are known for a long time and appear at the data interchange of CAD - NC data likewise. As a proper work-around, integrated CAD - FEM systems do exist which are only to acquire at a very high price.

Another attempt enlarges (better: blows up) the CAD system by e.g. additional modules or macros to such an extend, that partly utilizable FEA data can be produced. This is done frequently. It bears the disadvantage that it neither works well for all CAD programs nor works quite exactly even for the same products of one CAD program manufacturer.

Another attempt does nothing in the CAD system. The FEA system, however, contains a kind of mini- or semi-CAD system, in order to process or rework the raw and totally useless CAD data into FEA data, but only by massive support of the operator. The disadvantage is here, that the operator must master two CAD systems, and the integrated semi-CAD system has not got the performance and power of the real CAD system.

At Z88 these difficulties are solved as follows:

1: FROM CAD SYSTEM TO Z88:

1.1 in the CAD system:

Remark: This point case 1.1 will be explained in greater detail in chapter 2.7.2. This is a summary.

(1) Design your component. Order and layers as you like.

(2) Define the FEA structure or the super structure by lines and points. Any order and layers, therefore unproblematic and fast.

(3) Number the nodes with the TEXT function on the layer Z88KNR. Any order, therefore unproblematic and fast.

(4) Write the element information with the TEXT function on the layer Z88EIO. Any order, therefore unproblematic and fast.

(5) Outline each element with the LINE function on the layer Z88NET. The only section with firm work rules and orders (because of the topological informations).

(6) Write general information, material information and control information for the stress processor Z88D on the Layer Z88GEN.

(7) Define the boundary conditions on the layer Z88RBD.

(8) Export or store your 3-D model or 2-D drawing under the name Z88X.DXF.

1.2 in Z88: Starts the CAD converter Z88X

You can choose depending on your input data whether

** A mesh generator file Z88NI.TXT or*
** A file of the general structure data Z88I1.TXT or*
** A complete Z88 data set with Z88I1.TXT, Z88I2.TXT and Z88I3.TXT*

is produced. Everything else runs automatically.

1.3 in Z88: Starts other Z88 modules

Check output files produced by Z88X once more with the Filechecker Z88V.

Run the FEM analysis by starting the different Z88 modules at your choice:

** Mesh Generator Z88N*
** Plot Program Z88P*
** Direct Cholesky Solver Z88F*
** Iteration Solver Z88I1/Z88I2*
** Stress Processor Z88D*
** Nodal Force Processor Z88E*

2: FROM Z88 TO CAD PROGRAM

2.1 in Z88: Input files Z88xx.TXT

You have produced the input files

** Mesh generator file Z88NI.TXT or*
** File of the general structure data Z88I1.TXT or*
** complete Z88 data set with Z88I1.TXT, Z88I2.TXT and Z88I3.TXT*

either by an editor, a word processing program, EXCEL or an own routine or by modifying data files that came from the CAD converter Z88X.

2.2 in Z88: Launch CAD converter Z88X

Define which Z88 input files shall be converted. The DXF-file produced by Z88X is Z88X.DXF. If the input files contained polar- or cylindrical coordinates, they are converted into cartesian coordinates.

2.3 in the CAD system:

Import the DXF file Z88X.DXF. Save the loaded model or drawing under a valid CAD name (e.g. at AutoCAD name.DWG) and work with the drawing. You can switch off and switch on the different Z88-layers as you like.

2.7.2 Z88X IN DETAIL

Proceed in the following steps and reserve the following layers

Z88GEN: Layer for general information (1st input group in the mesh generator input file Z88NI.TXT and general structure data file Z88I1.TXT). Include further the material information (4th input group in the mesh generator input file Z88NI.TXT and general structure data file Z88I1.TXT). Add, if necessary, the data of the stress parameter Z88I3.TXT.

Z88KNR: Layer including the node numbers.

Z88EIO: Layer including the element information like element type and in the case of mesh generator input file Z88NI.TXT control information for the mesh generator.

Z88NET: Layer containing the mesh which was drawn or outlined in defined order.

Z88RBD: Layer containing the contents of the boundary conditions file Z88I2.TXT.

A further layer, **Z88PKT**, is produced by Z88X if you convert from Z88 to CAD. It shows all nodes with a point marker so that one better recognizes the nodes. For the reverse step, from CAD to Z88, it is completely insignificant.

1st step: Design your component in the CAD system as usual. You do not need to maintain a definite order and you can use any layers. It is highly recommended to put symbols on one layer, edges on another layer, dimensions on a third layer, invisible lines and center lines on a forth layer and so on. This enables you to remove all unnecessary information in the next step.

2nd step: Plan your mesh subdivision, that means suitable finite element types and their distribution. Subdivide the FE structure or the super structure into elements by lines, insert **all** points which are not yet existing (for example intersection points or end-points of lines are usable). Any order and layer. However, it is recommended not to use the Z88-layers like Z88NET, Z88GEN, Z88PKT, Z88KNR, Z88EIO and Z88RBD. Better define any new layer for this or use already available layers from step 1.

3rd step: Define the Z88-Layer Z88KNR and make it the active layer. Catch or trap every FE node, which were already defined in the 1st step by your construction or have been completed in the 2nd step, and number them. Write to every node **P blank node-number** e.g. *P 33*, with the TEXT function of the CAD program. Be very careful to snap exactly the node and attach the number exactly to the node's location. Take your time ! With the snap modes of AutoCAD (intersection point, end-point, point etc.) this works well. Choose any order of the work consequence as you like, you can well number the node 1 (*P 1*), then the node 99 (*P 99*) and then node 21 (*P21*). However, the numbering of the nodes must make sense and must be meaningfull for a FE analysis. *You* define which node in node 99 and which other node reads 21. Bad node numbering can cause heavy (but not really necessary) storage needs and computing times. Consult a good FEA book for this aspects.

4th step: Define the Layer Z88EIO and make it the active layer. Write the element information with the TEXT function anywhere (of course, it looks nicer with the element infos placed in middle of the respective finite element or super element). The order of the work consequence is up to you. You can describe element 1 first, step to the attaching element 17 and then proceed with element 8. However, your element choice and description must make sense for a FE analysis. The following information have to be written:

For all finite element types from 1 to 20 (not 16 and 17):

FE Element number Element type
Write into one line, separate each item by at least one blank.

Example: An Isoparametric Serendipity Plane Stress Element No.7 is supposed to get the element no. 23. Write e.g. into the middle of the element with the TEXT function *FE 23 7*

For super-elements 2-dimensional No. 7, 8, 11, 12 and 20

SE
Element number
Super-element type
Type of the finite elements to be produced by meshing
Subdivision in local x direction
Type of subdivision in local x direction
Subdivision in local y direction
Type of subdivision in local y direction
Write into one line, separate each item by at least one blank.

Example: Sudivide an Isoparametric Serendipity Plane Stress Element with 12 nodes (Element type 11) used as super-element into finite elements of type 7, i.e. Isoparametric Serendipity Plane Stress Elements with 8 nodes (Element type 7). Subdivide in local x direction three times equidistantly and subdivide in local y direction 5 times ascending geometrically. The super element is supposed to have the number 31. Write e.g. into the middle of the element with the TEXT function: *SE 31 11 7 3 E 5 L* (e or E for equidistant is equivalent)

For super-elements 3-dimensional Hexahedrons No.10

SE
Element number
Super-element type
Type of the finite elements to be produced by meshing
Subdivision in local x direction
Type of subdivision in local x direction
Subdivision in local y direction
Type of subdivision in local y direction
Subdivision in local z direction
Type of subdivision in local z direction
Write into one line, separate each item by at least one blank.

Example: Subdivide an Isoparametric Serendipity Hexahedron with 20 nodes (Element type 10) as super element into finite elements of the type Isoparametric Hexahedrons with 8 nodes (Element type 1). Subdivide equidistantly three times in local x direction, 5 times ascending geometrically in local y direction and subdivide equidistantly 4 times in local z direction. The super element is supposed to have the number 19. Write e.g. into the middle of the element with the TEXT function:
SE 19 10 1 3 E 5 L 4 E (e or E for equidistant is equivalent)

5th step: Define the Layer Z88NET and make it the active layer You need concentration for this step, because a firm and rigid work consequence must now be kept because of the topological information. One of the most important information, the coincidence, is defined in this step, that means which elements are defined or outlined by which nodes. Choose a proper color which differates well from the colors used till now and remove all superfluous information by switching off unused layers.
Select the LINE command and select the proper snap options e.g. points, intersection points and, if necessary, end-points.

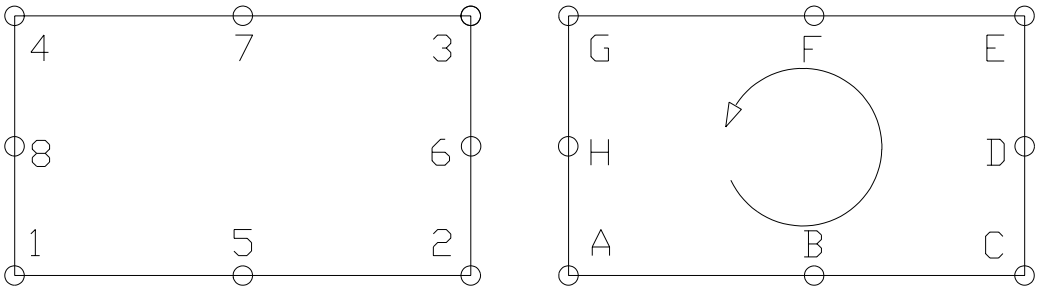
Start at the first element. For Z88 the first element is the element with which you start now, that means the one which you have chosen for your first element (*SE 1 or FE 1*). Select the node you want to be the first node of this element (this can be e.g. globally the node 150) and draw a line to the node which shall be the second node of this element (this can be e.g. globally the node 67). From there, draw a line to the third node of this element (this can be e.g. globally the node 45). Connect all required nodes with lines and draw at last a line to the starting point, the first node, then quit the LINE function.

Then you do the same with the second element. Remember: **You determine with this order which of the elements will be the real second element now**. In the previous 4th step you have only defined what kind of element the second element is. You determine here **how** the element is defined topologically.

The third element follows and so on. If you should make a mistake at the outlining of an element then delete all previous lines of this element (e.g. with an UNDO function) and start again at the first point of the questionable element. But if you notice now just outlining element 17 that you have made a mistake at element 9 , then you must delete all lines of the elements 9 to 17 and restart with element 9.

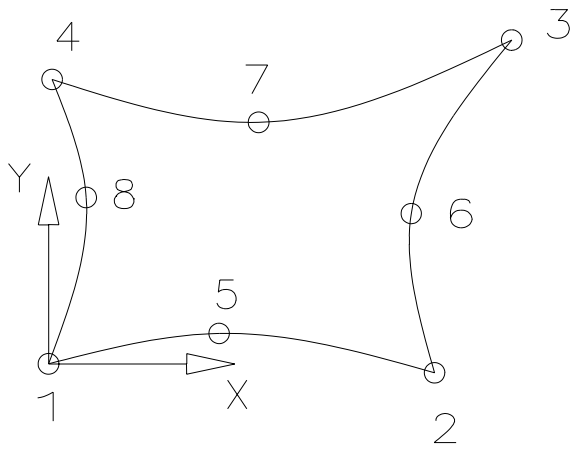
For your comfort, you must keep the following outline orders which partly differ from the orders shown at the element descriptions when entering the coincidence by hand. Z88X then sorts internally correctly.

Example: The coincidence for the element type 7 is as follows in the element description : First the corner nodes, then the middle nodes, reads 1-2-3-4-5-6-7-8. The coincidence list must look like this in the Z88 input files. However, for Z88X' use for comfortably outlining the elements the order is 1-5-2-6-3-7-4-8-1 (left picture) respectively A-B-C-D-E-F-G-H-A (right picture):

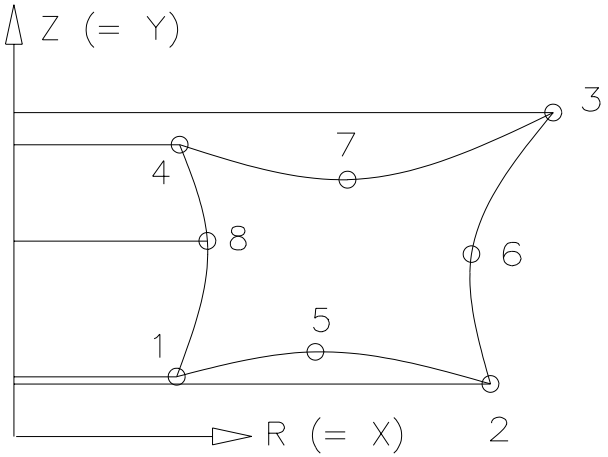


Following the CAD outline orders for all elements but No. 16 and No.17 (because these tetrahedrons can only machine- generated, nearly impossible by hand):

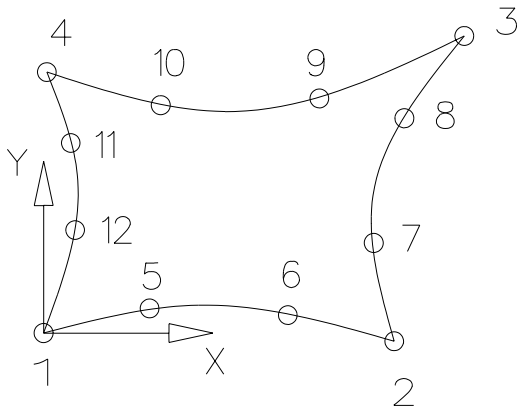
Element No.7 and No.20: 1 - 5 - 2 - 6 - 3 - 7 - 4 - 8 - 1



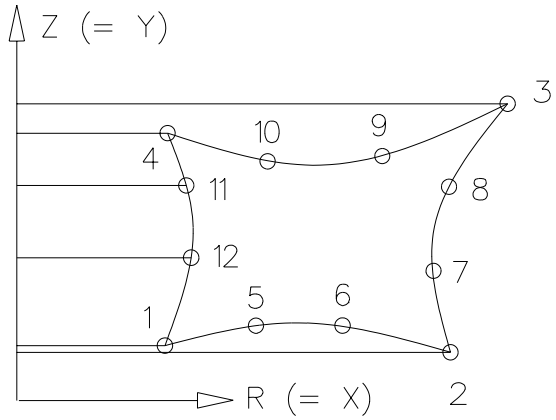
Element No.8: 1 - 5 - 2 - 6 - 3 - 7 - 4 - 8 - 1



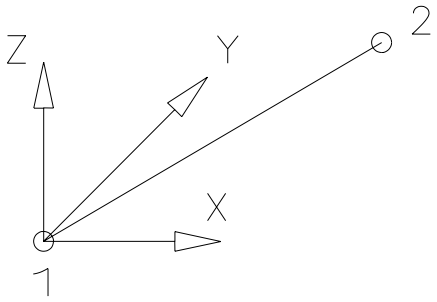
Element No.11: 1 - 5 - 6 - 2 - 7 - 8 - 3 - 9 - 10 - 4 - 11 - 12 - 1



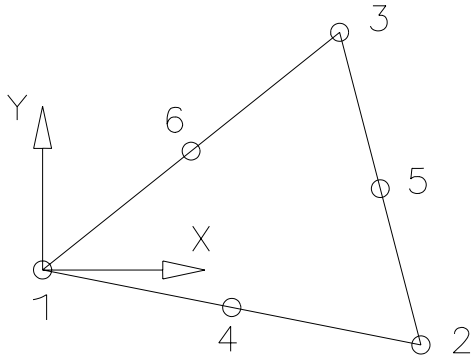
Element No.12: 1 - 5 - 6 - 2 - 7 - 8 - 3 - 9 - 10 - 4 - 11 - 12 - 1



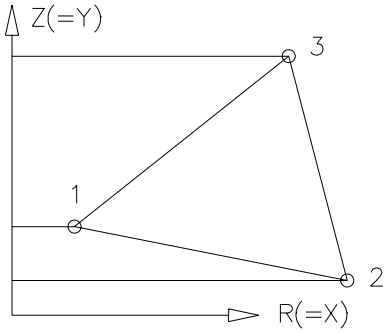
Element No. 2, 4, 5, 9, 13: Line from node 1 to node 2



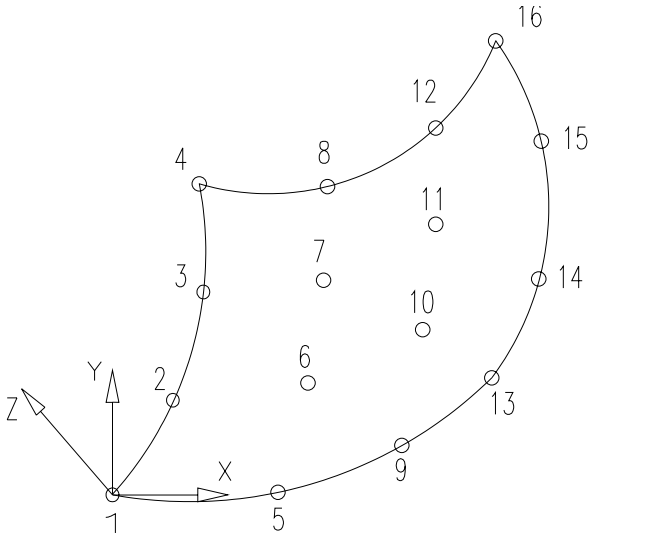
Element No.3, 14, 15 and 18: 1 - 4 - 2 - 5 - 3 - 6 - 1

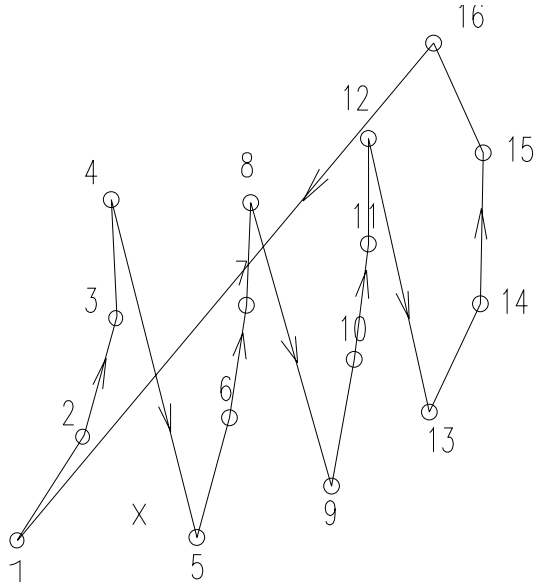


Element No.6: 1 - 2 - 3 - 1

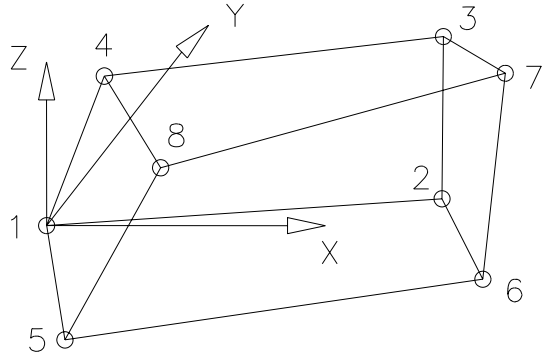


Element No.19: 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 -10 - 11 - 12 - 13 - 14 - 15 - 16 - 1



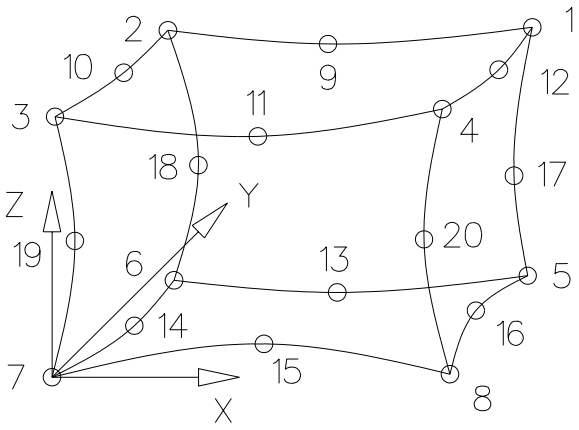


Element No.1:
Upper plane: 1 - 2 - 3 - 4 - 1, quit LINE function
Lower plane: 5 - 6 -7 - 8 - 5, quit LINE function
1 - 5, quit LINE function
2 - 6, quit LINE function
3 - 7, quit LINE function
4 - 8, quit LINE function



Element No.10:

Upper plane: 1 - 9 - 2 - 10 - 3 - 11 - 4 -12 - 1, quit LINE function
Lower plane: 5 - 13 - 6 - 14 - 7 - 15 - 8 - 16 - 5, quit LINE function
1 - 17 - 5, quit LINE function
2 - 18 - 6, quit LINE function
3 - 19 - 7, quit LINE function
4 - 20 - 8, quit LINE function



6th step: Define the layer Z88GEN and switch it active. Write with the TEXT function into a free space (well into any place of your drawing):

6.1 general information, i.e. the first input group of the general structure data Z88I1.TXT or the mesh generator file Z88NI.TXT,
In case of Z88I1.TXT (i.e. FE mesh):

Z88I1.TXT
Dimension of the structure
Number of nodes
Number of finite elements
Number of degrees of freedom DOF
Number of material information lines
Coordinate flag (0 or 1)
Beam flag (0 or 1)
Plate flag (0 or 1)

Write into one line, separate each item by at least one blank. **Definitely write in the layer Z88GEN.**

Example: 3-dimensional FE structure with 150 nodes, 89 finite elements, 450 degrees of freedom, 5 material information lines. Input with cartesian coordinates, structure contains neither beams No.2 nor beams No.13. Thus *Z88I1.TXT 3 150 89 450 5 0 0*

In case of Z88NI.TXT (i.e. super structure):

Z88NI.TXT
Dimension of the structure
Number of nodes
Number of super element
Number of degrees of freedom DOF

Number of material information lines
Coordinate flag (0 or 1)
Beam flag (must here be 0!)
Plate flag (0 or 1)
Trap radius header flag (most 0)

Write into one line, separate each item by at least one blank. **Definitely write in the layer Z88GEN.**
Example: 2-dimensional super-structure with 37 nodes, 7 super elements, 74 degrees of freedom, one material information line. Cartesian coordinates, no beams (anyway forbidden in the mesh generator file), no plates, use default for trap radius. Thus
Z88NI.TXT 2 37 7 74 1 0 0 0 0

6.2 Material information lines:

For every material information one separate line:

MAT
Number of the material information
This material information starts with element no. abc inclusively
This material information ends with element no. xyz inclusively
Young's Modulus
Poisson's Ratio
Integration order (from 1 to 4)
Cross section value (e.g. for plane stress elements thickness, for trusses cross section area)

... And if beams (but not plates !) are defined in addition:

Second moment of inertia yy (bending around yy axis)
Max. distance from neutral axis yy
Second moment of inertia zz (bending around zz axis)
Max. distance from neutral axis zz
Second moment of area (torsion)
Second modulus (torsion)

... And if plates (but not beams !) are defined in addition:

area load

Write into one line, separate each item by at least one blank. **Make sure to write in the layer Z88GEN.**

Example: The structure has 34 super elements type 7 with varying thickness: Elements 1 to 11 have thickness 10 mm, elements 12 to 28 have 15 mm and elements 29 to 34 have 18 mm. Material steel. Integration order shall be 2.

```
MAT 1   1   11   206000.  0.3  2  10.
MAT 2  12   28   206000.  0.3  2  15.
MAT 3  29   34   206000.  0.3  2  18.
```

6.3 Stress parameters:

The input line of the stress parameter file Z88I3.TXT

Z88I3.TXT
Integration order (0 to 4)
KFLAG (0 or 1)
Von Mises stresses (0 or 1)

Write into one line, separate each item by at least one blank. **Make sure to write in the layer Z88GEN.**

Example: The structure uses finite elements type 7. The stress calculation is supposed to be carried out in 3*3 Gauss points per element, stresses are supposed to be calculated in addition radially and tangentially. Compute von Mises stresses, too. Thus *Z88I3.TXT 3 1 1*

7th step: Define the Layer Z88RBD and activate it. Write with the TEXT function into a free space (well into any place of your drawing):

7.1 number of the boundary conditions, i.e. the first input group of the boundary condition file Z88I2.TXT

Z88I2.TXT Number of the boundary conditions

Write into one line, separate each item by at least one blank. **Make sure to write in the layer Z88RBD.**

Example: The structure has 10 boundary conditions, e.g. two loads and eight constraints i.e. support reactions. Thus *Z88I2.TXT 10*

7.2 Boundary conditions, the second input group of the boundary condition file Z88I2.TXT

RBD
Number of the boundary condition
node number
Degree of freedom
Header flag force/displacement (1 or 2)
Value

Write into one line, separate each item by at least one blank. **Make sure to write in the layer Z88RBD.**

Example: The structure shall be a truss- framework. Node 1 shall be fixed in Y and Z, node 2 fixed in X and Z. Nodes 7 and 8 have a load of 30,000 N each in Z direction, pointing down. Node 19 is fixed in X and Z and node 20 is fixed in Y and Z. Thus

RBD 1 1 2 2 0
RBD 2 1 3 2 0
RBD 3 2 1 2 0
RBD 4 2 3 2 0
RBD 5 7 3 1 -30000
RBD 6 8 3 1 -30000
RBD 7 19 1 2 0
RBD 8 19 3 2 0

RBD 9 20 2 2 0
RBD 10 20 3 2 0

8th step: Export (store) your model or drawing under the name Z88X.DXF in the DXF file format. For precision of decimal positions take the default value which the CAD program suggests. Take care that you export directly into the Z88-directory or you must copy the file Z88X.DXF by hand into the Z88-directory, because the CAD converter Z88X expects the input and output files in the same directory, where Z88X is located.
You may launch the CAD converter Z88X then.

Note: If you want to convert Z88 text files as Z88X.DXF to CAD, you can choose the text size which applies to all texts like node numbers, element numbers etc. This is very important from time to time because there is no possibility in e.g. AutoCAD to change the text size globally afterward. From time to time you must make some trys untill you have found the suitable text size for the respective Z88 file. Simply call Z88X once more with another text size.

Windows: *In Z88X: File > Textsize*
UNIX: *z88x -i1tx | -iatx | -nitx | -i1fx | -iafx | -nifx -ts number*

Caution, valuable note: Use the Z88X keywords **"P number, FE values, SE values, MAT, RBD, Z88NI.TXT, Z88I1.TXT, Z88I2.TXT and Z88I3.TXT"** only where they are really needed. Take care that they do not appear in other drawing items ! Otherwise Z88X cannot interpret the DXF file properly and will flag error messages !

2.8 THE COSMOS CONVERTER Z88G

Sometimes 3D CAD programs include so-called automeshers which devide a CAD model into finite elements. This generated mesh can be stored in some output format to fit the needs of the various FEA programs.

Typical output formats are the COSMOS and the NASTRAN format for the *COSMOS* or the *NASTRAN* FEA program.

Z88G is developed and tested for *Pro/ENGINEER* by Parametric Technology. USA. Pro/ENGINEER must include the option *Pro/MECHANICA*. Be sure to define the material data (e.g. for steel, only Young's Modulus and Poisson's Ratio is really needed) in Pro/ENGINEER.

Then you may activate *FEM* in the Pro/ENGINEER program after designing your 3D model , define a coordinate system (which must be in harmony with Z88 !) and add forces and boundary conditions to single points. Create these single points with *Feature* > *Datum* > *Point*.

Modify the mesh control values, if necessary. Create the mesh with *Make Model* and choose the element type e.g. *Tet Mesh* or *Shell Mesh*. Store the mesh with *Output Model*, choose *NASTRAN* or *COSMOS/M* and *linear* or *parabolic*. Enter *z88g.nas* for NASTRAN files or *z88g.cos* for COSMOS files for the output file name.

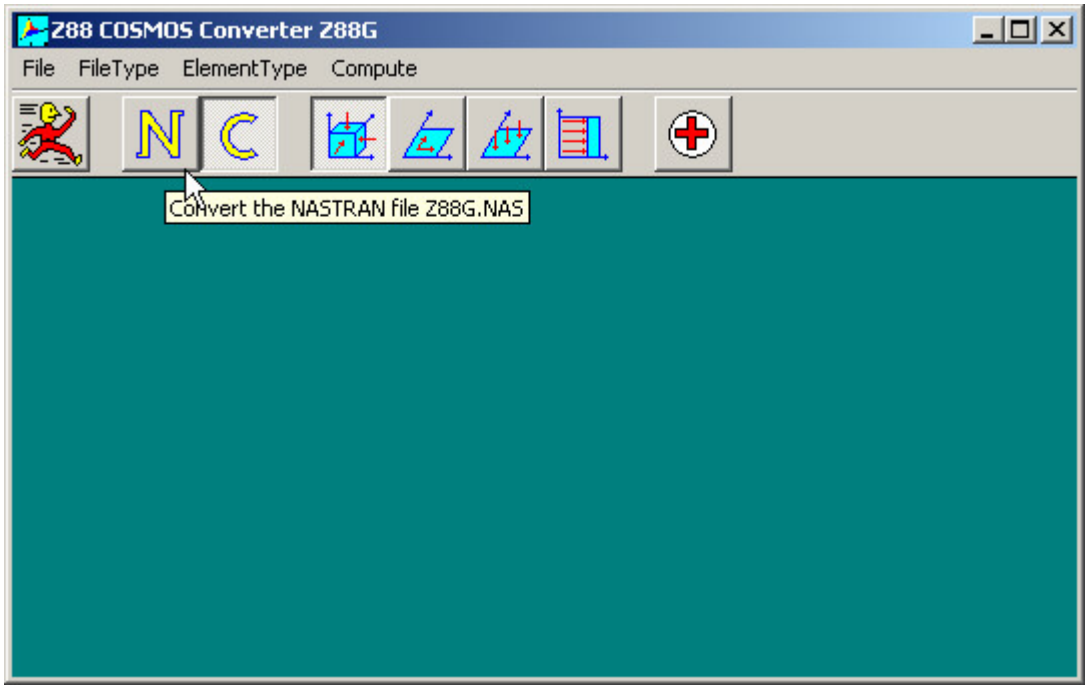
Then lauch the converter **Z88G**. The converter produces the Z88 input files Z88I1.TXT, Z88I2.TXT and Z88I3.TXT automatically. You may then enter the Z88 input files and edit values e.g. material data and integration orders, if necessary.

Test the Z88 input files generated by Z88G with the filechecker Z88V. Plot Z88I1.TXT with the plot program Z88O or Z88P. If you find a 3D model totally flat: You've defined a coordinate system CS0 in Pro/ENGINEER which does not fit Z88's needs. Simply define a new correct coordinate system in Pro/ENGINEER and define it as datum when outputting the model.

You may create the following Z88 element types with Z88G:

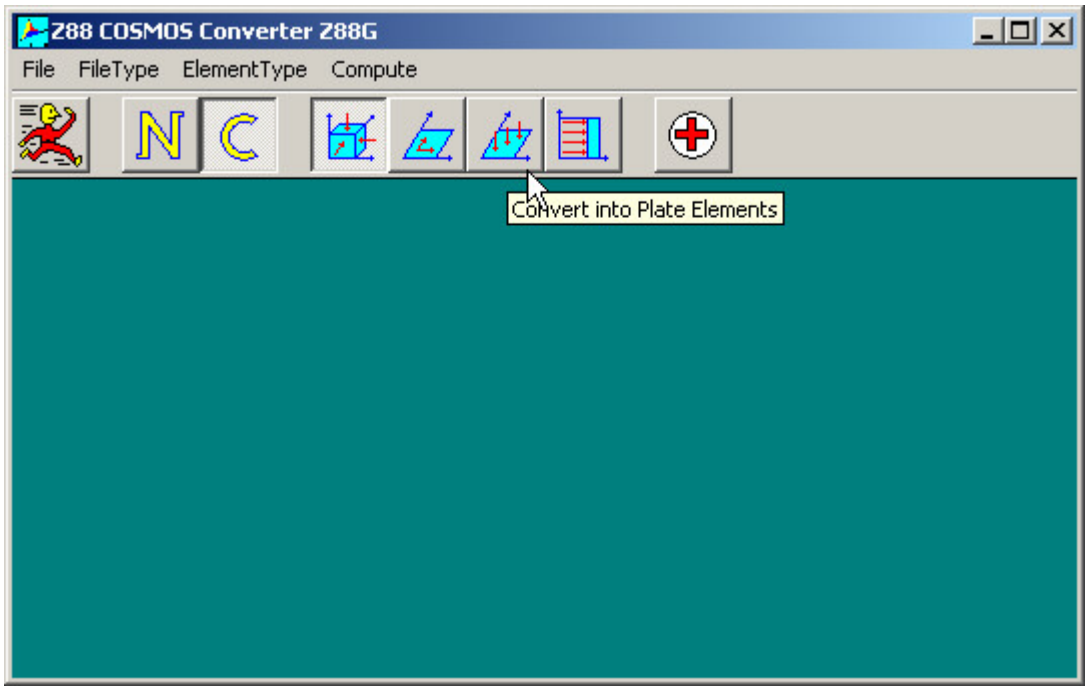
Tetrahedron No.16	(*Tetrahedron*, *parabolic* in Pro/ENGINEER)
Tetrahedron No.17	(*Tetrahedron*, *linear* in Pro/ENGINEER)
Plane stress No.14	(*Shell, triangle, parabolic* in Pro/ENGINEER)
Plane stress No.7	(*Shell,quadrangle, parabolic* in Pro/ENGINEER)
Plate No.18	(*Shell, triangle, parabolic* in Pro/ENGINEER)
Plate No.20	(*Shell,quadrangle, parabolic* in Pro/ENGINEER)
Torus No.15	(*Shell, triangle, parabolic* in Pro/ENGINEER)
Torus No.8	(*Shell,quadrangle, parabolic* in Pro/ENGINEER)

How to proceed?
First step: Choose NASTRAN or COSMOS file format: If you choose NASTRAN the file Z88G.NAS is loaded, in case of COSMOS the file Z88G.COS is loaded. You must know which file type did you file in your former Pro/E session.



(Choose file type before start. The UNIX version of Z88G operates in console mode)

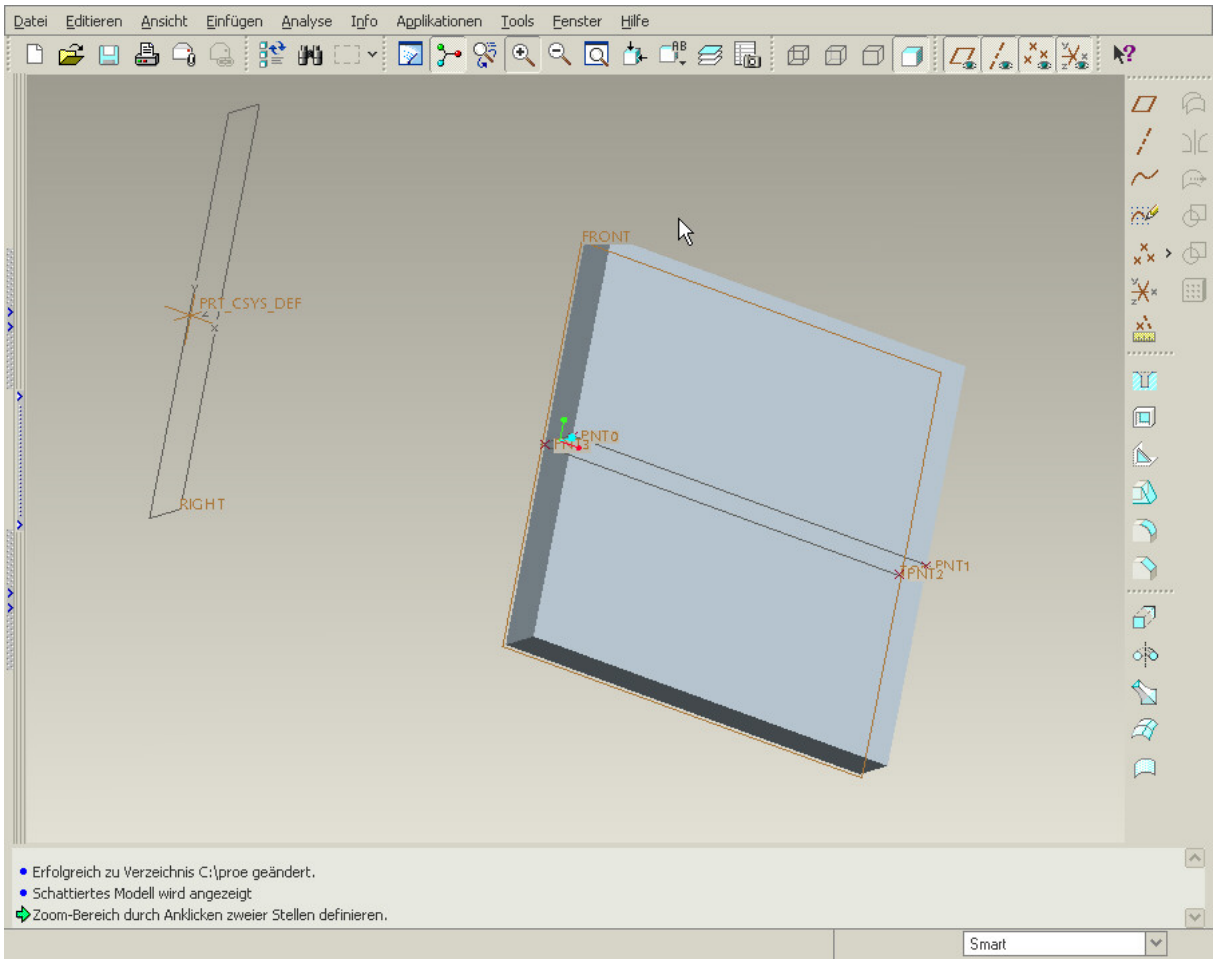
Next step: Pro/ENGINEER makes no distinction between plane stress elements, torus elements and plate elements, so, it's up to you to feed Z88G with the right information; choose the proper element type (the type you prepared in your former Pro/E session) in Z88G before starting the conversation run:



(Before running the conversation choose the right type of elements; works similar for the UNIX version of Z88G in console mode)

The generation of volumes is easy but the generation of plane stress elements, plates and torus elements is tricky: Firstly, build a volume with (small) thickness in Pro/E. Set reference points, especially for axisymmetric elements. Launch Pro/MECHANICA and *idealize* the volume into shells: *Model* > *Idealizations* > *Shells* > *Midsurfaces*. This eliminates the depth. When working with axisymmetric elements keep in mind that you are working in cylinder

coordinates: Your coordinate system coincidates with the axis of rotation and the “volume” lies on the corresponding radiuses:



(Here you see the generation of torus elements in Pro/ENGINEER (Wildfire). Proceed similar for plane stress and plate elements)

Please keep in mind: These FEA output data formats, especially the NASTRAN format, are really monthly modified. However, why should they kept for a while in the same state? Would be too easy. The COSMOS format is more stable but is missing in *Pro/ENGINEER Wildfire*. Thus, if you’re using *Pro/E* up to version *2001* you should store COSMOS files and starting from *Wildfire* store NASTRAN files. If you’ll store NASTRAN files for Pro/E versions up to 2001 you must check and modify the NASTRAN files: especially material properties are printed as wrong floating point values e.g. *2.06+5*. Change such stuff to e.g. *2.06E+5* or *206000*.

Anyway: Z88G looks quite harmless, but proper operated Z88G is a mighty tool which allows you to file very large FEA structures to Z88.

2.9 THE CUTHILL- McKEE PROGRAM Z88H

The choice of the nodal numbers is extremely important for the compilation of the stiffness matrix and bad nodal numbering may result in huge memory needs which are not really necessary.

However, Z88H may reduce the memory needs for the direct Cholesky Solver Z88F greatly. The iteration solver Z88I1/Z88I2 may also gain some advantages from a Z88H run, but the iteration solver is a-priori very stable regarding node numbering because of storing the non-zero elements only.

Basically, it it always good to achieve a small difference of nodal numbers for each finite element. This results in nodal numbers of similar size for an element. However, this is not always possible: consider a circular structur starting with nodal numbering at 0° with increasing numbers clockwise. When reaching 360°, elements with large differences of nodal numbers will occur.

Sometimes 3D CAD programs include so-called automeshers which devide a CAD model into finite elements. This generated mesh can be stored in some output format to fit the needs of the various FEA programs. But many of these automeshers generate meshes with very large nodal differences. This is true for Pro/ENGINEER's Pro/MECHANICA: If you choose *Tet Mesh parabolic*, Pro/MECHANICA in a first operation generates linear tetrahedrons, i.e. with 4 rather than 10 nodes per element, with straight element edges. Then midnodes are put on the element edges resulting in parabolic elements with 10 nodes. These midnodes have relatively large nodal numbers because the corner nodes were numbered in the first step. Thus, every finite element features relatively small corner node numbers and relatively large mid node numbers resulting in large differences of nodal numbering. When choosing *Shell, triangle, parabolic,* the same situation occurs. This means that meshes built with Pro/MECHANICA will always have bad nodal numbering.

For large meshes one needs to re-number the nodes to get finite elements with small differences of nodal numbers. Several proper procedures do exist in literature for this task. However, the so-called *Cuthill- McKee* procedure is a good compromise. One modification of it is the *reverse Cuthill- McKee algorithm*. For more information, consult *Schwarz, H.R.: Die Methode der finiten Elemente.* The C program Z88H is based on a FORTRAN77 program of Prof. Schwarz and is specially adapted to Z88. The core algorithm of H.R. Schwarz decides internally wether to use the normal *Cuthill- McKee procedure* or the *reverse Cuthill- McKee algorithm.*

The Cuthill- McKee program Z88H was originally designed for finite element meshes generated by COSMOS converter Z88G. However, Z88H can deal with all Z88 meshes. Z88H reads the Z88 input files Z88I1.TXT (general structure informations) and Z88I2.TXT (boundary conditions), files backups Z88I1.OLD and Z88I2.OLD and computes the modified input files Z88I1.TXT and Z88I2.TXT.

Own research studies showed that sometimes a second run of Z88H may improve again the numbering of a first run of Z88H. A third run seems to make things worse. In contrast, Z88H may sometimes compute a worse nodal numbering then the original mesh. You should have some experiments because the *Cuthill- McKee algorithm* may not always improve a given mesh.

And here's how you proceed:

1) Generate a finite elements mesh, i.e. the Z88 input files Z88I1.TXT and Z88I2.TXT. This can be done by:

- hand
- Z88 net generator Z88N (Z88I1.TXT only, then edit Z88I2.TXT by hand)
- a DXF file and Z88X
- a COSMOS file and Z88G

2) Adjust Z88.DYN if necessary: MAXKOI is very important (Number of nodes per element * total number of elements) and MAXK, MAXE and MAXNFG.

3) Launch Z88F with test option, i.e.

Windows: Z88F > Mode > Test Mode, Compute > Go
UNIX: z88f -t (console) or Z88F with option Test (Z88COM)

Fix the value for GS, i.e. the number of storage entries in the stiffness matrix (multiplying this values by 8 gives the memory need in bytes).

4) Launch Z88H.

5) Repeat step 3, i.e. run Z88F with the test option and check wether GS got smaller. This will be mostly the case if your mesh was generated by Z88G using a COSMOS file. Otherwise, restore Z88I1.TXT and Z88I2.TXT from the backup files Z88I1.OLD and Z88I2.OLD.

6) Enter the value of GS into Z88.DYN in the line MAXGS and lauch Z88F with a compute mode, e.g.

Windows: Z88F > Mode > Compact Mode, Compute > Go
UNIX: z88f -c (console) or Z88F with option Compact (Z88COM)

Remark:

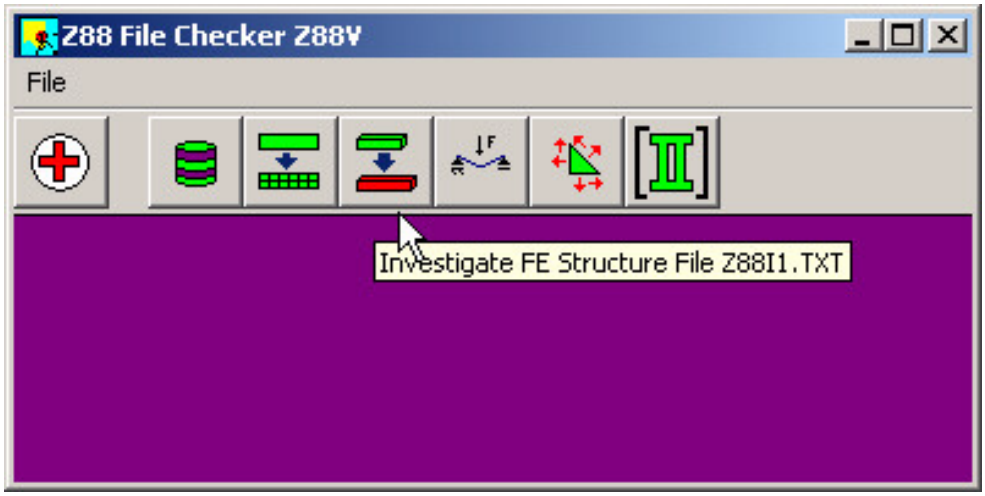
Z88H features a section in the memory header file Z88.DYN:

CUTKEE START
MAXGRA 200 (maximum degree of nodes)
MAXNDL 1000 (steps of the algorithm)
CUTKEE END

Increase these entries for very large structures.

2.10 THE FILECHECKER Z88V

This program investigates the Z88 input files Z88I1.TXT, Z88I2.TXT, (both for the FE processor Z88F), Z88I3.TXT (header file for the stress processor Z88D), Z88I4.TXT (header file for the iteration solver Z88I1/Z88I2) as well as the net generator input file Z88NI.TXT (for the net generator Z88N) for typing mistakes and logical faults. Cross-checks are executed, i. e. Z88I2.TXTund Z88I3.TXT are only examined after Z88I1.TXT has been checked properly.



(Windows Z88V. On UNIX, Z88V will run in text mode)

Although Z88V recognizes many conceivable fault possibilities and is internally quite tricky, situations like with compilers may occur where faults are not detected or seem to be recognized on other passages. Z88V distinguishes between warnings and errors. At warnings Z88V continues directly or asks for continuing. Z88V stops when detecting the first error because otherwise resulting sequence errors are usually generated from this. Therefore, a recognized error must be fixed right now.

An error-free input file recognized from Z88V can nevertheless lead to subtle faults at the later program run. However, the probability is low to some extent. This statement refers to formal errors: Z88V neither recognizes inconsistent structures, nor wrong or too few boundary conditions !

Note:
Always check FE calculations with analytical rough calculations, results of experiments, plausibility considerations and other checks without exception !

2.11 THE OPENGL PLOT PROGRAM Z88O

With the new OpenGL plot program Z88O the Z88 system enters new dimensions. You may illuminate a structure with three different light sources or plot with hidden lines, both the undeflected and the deflected structure. You may plot stresses and X, Y and Z deflections with a color range - like the expensive professional FEA programs. You may plot a limited range of nodal or element numbers - a nice feature especially for large structures. A printer or plotter feature is not included into Z88O - and why - just do a screen shot.

Z88O uses OpenGL so your computer must be able to deal with OpenGL graphics. This is true for all newer Windows machines and a quite cheap graphics card will do well. Anyway, it's always a good idea to control the system settings - sometimes you may turn on OpenGL hardware acceleration. Compilation (only necessary if you want to add improvements) under Windows is easy because OpenGL is part of Windows since Windows95 and the well-known compilers (MS Visual C++, LCC, Borland C++ Builder, OpenWatcom) come with the necessary OpenGL libraries and header files.

Professional UNIX workstations have always OpenGL features included. If you want to compile the Z88 system by yourself on a UNIX machine be sure to have the libraries *libGLw* and *libGL* and the header files *gl.h, glx.h* und *GlwMDrawA.h* on your machine in the proper library and include paths. This is also true for LINUX systems, but LINUX sometimes presents problems with OpenGL and your graphics card. If so, don't blame Z88O because Z88O was developed on a SGI. And SGI makes the OpenGL reference machines. But with the newer RedHat and SuSE distributions it should work after some effort. If Z88O won't work at all: The approved plot program Z88P which works with the WinAPI or the X11 libraries stays in Z88, ref. Chapter 2.6.

Of course, you may define your choice of colors, the light features, material properties, the polygon offset and the fonts (for UNIX) in the parameter files Z88O.OGL (for Windows) and Z88.FCD (for UNIX). The file Z88.FCD for the UNIX operation system allows for much more possibilities in changing colors and fonts as well as the sizes and locations of the pushbuttons, radioboxes etc. You can completely modify the whole appearance of Z88O for UNIX as you wish. Be careful with changes in Z88O.OGL (Windows) or Z88.FCD (UNIX/LINUX): You should have some proper knowledge about OpenGL if you want to change light effects etc. Otherwise you may pull a long face because nothing will work as you wish. Some hints are included into Z88O.OGL and Z88.FCD, however, i can not give here an introduction into OpenGL. Consult the two basic books „OpenGL Programming Guide" and „OpenGL Reference Manual" from Addison-Wesley.

Start of Rendering: When Z88O was launched the OpenGL subsystem is started and prepared to go. You'll start rendering under Windows with the very left icon *Go* and under UNIX with the pushbutton upper row right <*Run*>.

Needed Files	*Super structures*	*Undeflected FE struc.*	*Deflected FE struct.*
Z88NI.TXT	Yes	No	No
Z88I1.TXT	No	Yes	Yes
Z88O2.TXT	No	No	Yes
Z88O8.TXT	No	Yes, for Mises stresses	Yes, for Mises stresses

Rendering with Z88O: For fastest operation Z88O connects the nodal points - and only the corner points- with straight lines, although for Serendipity elements the edges of the elements are square or cubic curves. However, especially illuminated scenes need a huge amount of computational power. Please keep in mind: If a part renders pretty fast in your CAD system, Pro/ENGNEER for example, and the same part renders quite slowly in Z88O - this is normal business because CAD systems are „drawing“ only some outline curves. In contrast, FEA system have to render *every* finite element i.e. compute the normal vectors for any element surface, compute light effects for every tetrahedron etc. Hidden line scenes put very heavy load on the CPU, too.

What can i plot with Z88O? Nearly everything if a solver (Z88F or Z88I1 with Z88I2) was run which stored the deflection file Z88O2.TXT along with a run of the stress processor Z88D which stored the three stress files Z88O3.TXT (for you to check the stresses), Z88O5.TXT (for Z88P) and Z88O8.TXT (for Z88O). Even for trusses you may plot the „von Mises“ stresses (i.e. tensile stresses) with different colors. Only beams No.2 and No.13 and cams No.5 allow only the plotting of deflections and nothing more. Why? Because you must compute for beams and cams also the notch sensitivity which is impossible for a FEA system which deals with a *whole structure of beams*.

Plot of stresses: The kind of plotting the stresses within FEA programs is truly of philosophical character. As a matter of fact, numerous experiments and computer studies at the *Institute of Engineering Design and CAD* of the University of Bayreuth, Germany, showed, that some very expensive and well-known professional FEA programs produced *incorrect stress plots* in some situations! The best way is the computation of stresses directly in the Gauss points as realized in Z88P. However, this is odd for OpenGL so i decided for the following way after a lot of experiments:

- *von Mises stresses in corner nodes.* In fact, the stresses are computed not really in the corner nodes which would lead to very wrong results especially for very tapered elements but in Gauss points laying in the near of the current corner nodes. Stresses are computed for just the same number of Gauss points like the number of corner points. Because often a node is linked to more than one element the stresses are computed to a mean value from the „corner node“ stresses of all linked elements. This results in pretty balanced stress shadings which are mostly somewhat lower than the maximum stresses of Z88P, however. The value of the order of integration INTORD in the header file Z88I3.TXT has no meaning but INTORD should be greater than 0.
- *von Mises stresses as a mean value for each element.* The stresses are computed in the Gauss points of the current element, added and then divided by the current number of Gauss points. This results in a mean value for the *von Mises stress* per element. The value of the order of integration INTORD in the header file Z88I3.TXT is important and INTORD must be greater than 0.

Plot of deflections: You may plot the undeflected or the deflected structure. The enlargement factor is adjustable, with 100 as the default value for X, Y and Z. In addition, you may plot the deflections for X, for Y or for Z with color shading. This is a pretty nice feature for large spatial structures. In contrast to Z88P, you may plot the shaded colors for stresses or for the deflections with the deflected structure, too.

Hints for the user for Zooming, Panning and Rotating:

1. You may work without limitation with the special keys for Windows (see below) or the pushbuttons for UNIX. You should use the special keys or the pushbuttons for precise zooming, panning and rotating. This is the default mode. Mouse navigation is turned off.

2. With Z88O – in contrast to Z88P- you may use mouse navigation: Under Windows, press the mouse icon. Under UNIX, press the pushbutton *Pushbu* (which changes to *Mouse*): Now you can
    - zoom with the left mouse button pressed
    - pan with the middle mouse button pressed
    - rotate with the right mouse button pressed

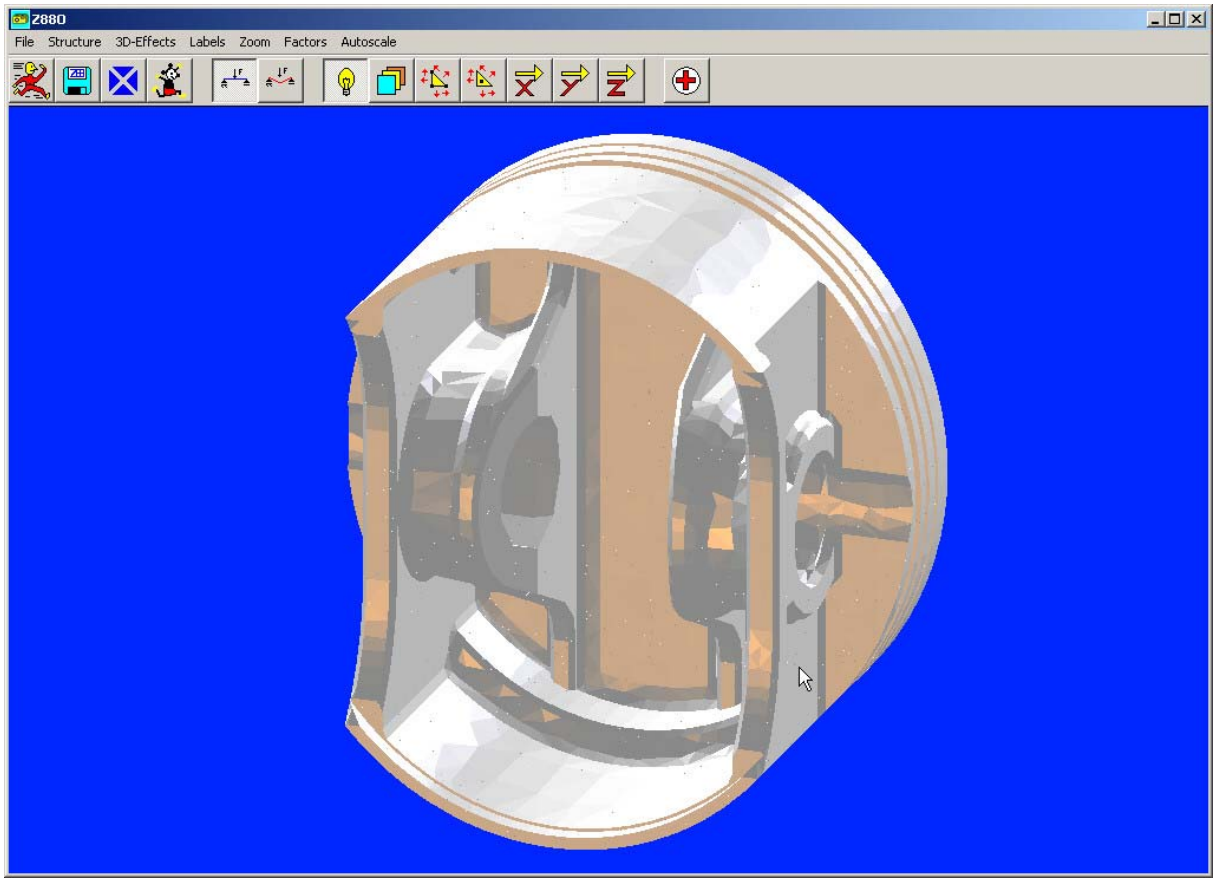
This option fits well for limited zooming- and panning ranges and for fast but quite unprecise rotating. You may in addition use the special keys or pushbuttons but this mixed mode is not a real feature and may lead to unpredictable results because Z88O uses different calculations for both modes.

Special key strokes for Windows:

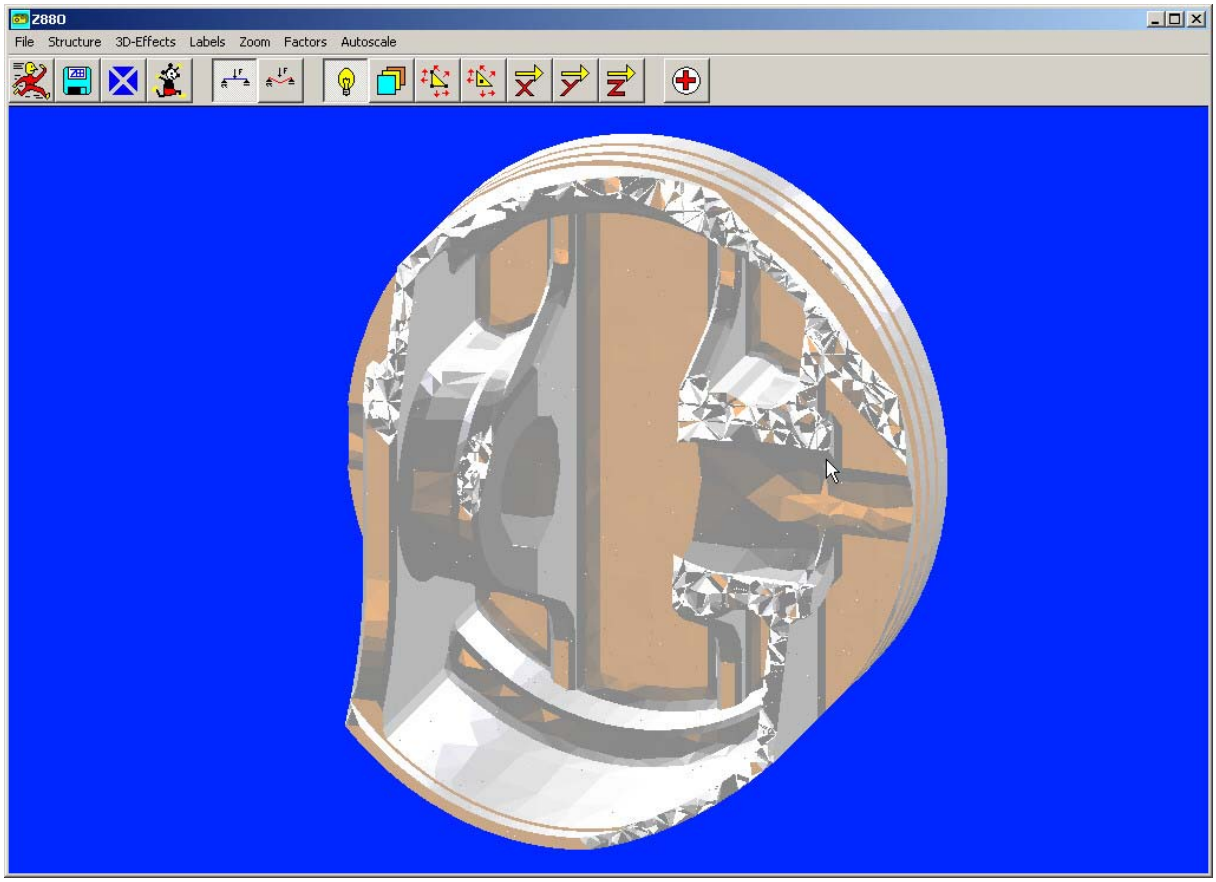
Prior:	increase zoom
Next:	decrease zoom
Cursor left:	panning X direction
Cursor right:	panning X direction
Cursor up:	panning Y direction
Cursor down:	panning Y direction
Home:	panning Z direction
End:	panning Z direction
F2:	rotate around X axis
F3:	rotate around X axis
F4:	rotate around Y axis
F5:	rotate around Y axis
F6:	rotate around Z axis
F7:	rotate around Z axis
F8:	reset all rotations to 0

Under UNIX use the usual X and Motif key assignments: Tab key and arrow keys for choice and space bar for activation.

The „coordinate system“: OpenGL works with a *Clipping Volume*, i.e. with a kind of cube, defined by *Xmin* and *Xmax* in horizontal direction, by *Ymin* and *Ymax* in vertical direction and *Zmin* (points towards the user) and *Zmax* (points away from the user). If you use a too-large zoom factor or if you are panning the structure too near to you then the range of *Zmin* is exceeded and parts of the structure are laying outside the viewing volume. This offers a nice chance to look into a structure. Otherwise, change the value of Z*min* (default entry is –100) to lower values, e.g. -1000: under Windows use *Factors > Z limit towards you*, under UNIX change the textfield „Zlimit“ right side below. The following screenshots are showing the situation:



Windows: piston of a BMW engine (motorcycle F650GS) Zlimit: default value –100.



Windows: piston of a BMW engine (motorcycle F650GS) Zlimit is –10, piston has slash cut.

Explanation of some menu items:

Name of Structure File:

Windows: *File > Structure File*
UNIX: *Stru.* text field directly on window

Choose the structure file here. Enter name, if necessary with path, press return. The new stucture is loaded. You'll start rendering under Windows with the very left icon *Go* and under UNIX with the pushbutton upper row right <*Run*>.

Deformation Modes of the Structure:

Windows: *Structure > Undeflected, Deflected*
UNIX: Radio box *Undefle., Deflected*

Plots the undeflected structure or the deflected structure. You may do all other rendering operations with the undeflected structure or the deflected structure.
Caution Deflected: The user must have executed a calculation of displacements before using this function. Do a FEA run with Z88F or Z88I1/Z88I2 before using Z88O. Otherwise, some old files Z88O2.TXT (displacements) from earlier Z88 runs are opened causing totally wrong results !!

Choice of the 3D effects:

Windows: *3D-Effekte >*
1. Light on,
2. Hidden Line on,
3. Mises stresses (corners),
4. Mises stresses (elements),
5. X Displacements on,
6. Y Displacements on,
7. Z *Displacements* on
or the appropriate icons

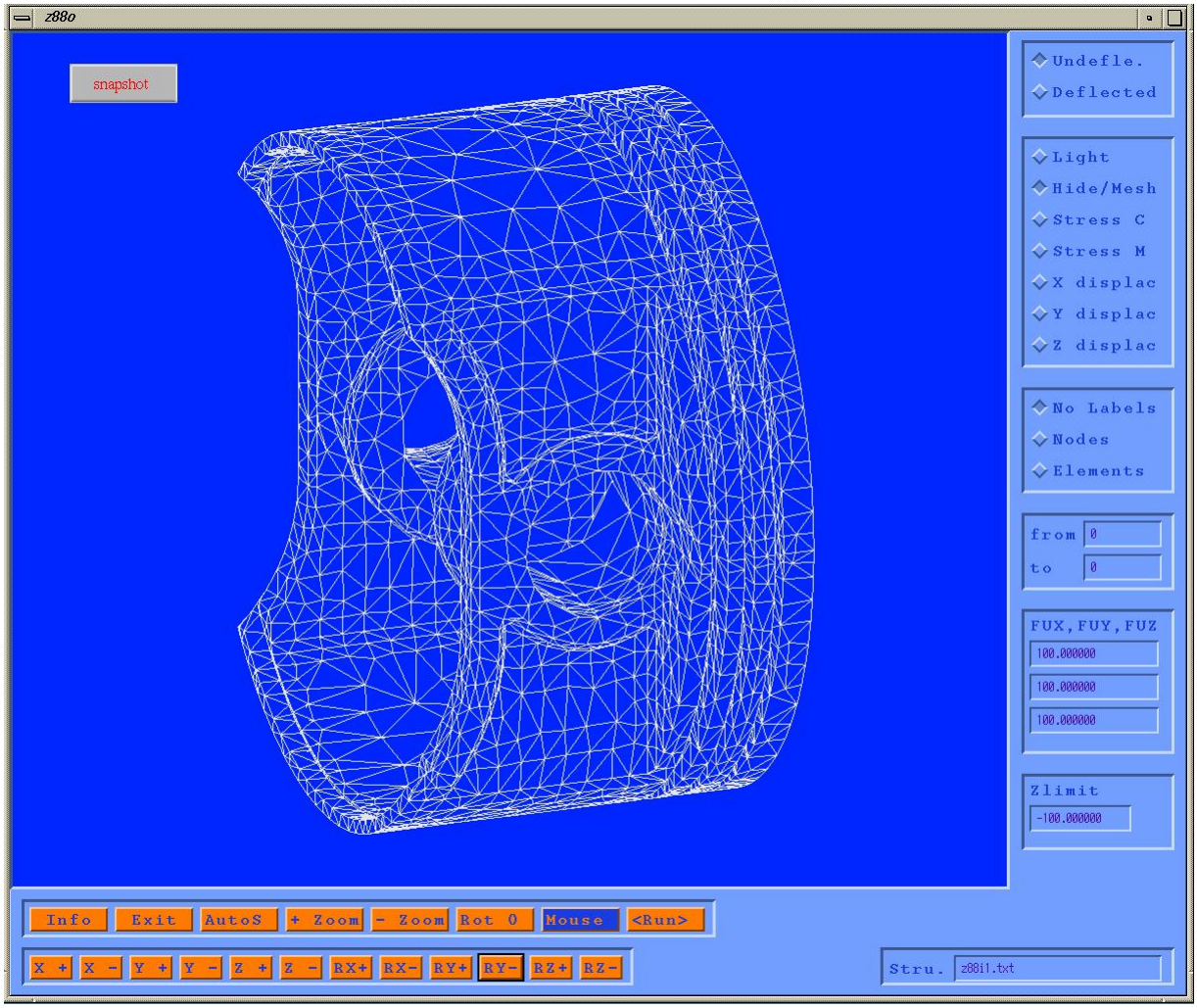
UNIX : Radiobox
1. Light
2. Hide/Mesh,
3. Stress E,
4. Stress M,
5. X displac,
6. Y displac,
7. Z displac

1. The structure is illuminated with three light sources. You may modify the features of the light sources by editing the header files Z88O.OGL (Windows) and Z88.FCD (UNIX).
2. For spatial structures the finite elements mesh is rendered with hidden lines. For 2D struktures the pure finite elements mesh is drawn (there is nothing to hide). Only in this mode you can see *all* desired nodal and elements labels. The polygon offset can be edited in the header files Z88O.OGL (Windows) and Z88.FCD (UNIX).
3. The *von Mises* stresses of the corner nodes are plotted. In fact, the stresses are computed not really in the corner nodes which would lead to very wrong results especially for very tapered elements but in Gauss points laying in the near of the current corner nodes. Stresses are computed for just the same number of Gauss points like the number of corner

points. Because often a node is linked to more than one element the stresses are computed to a mean value from the „corner node“ stresses of all linked elements. This results in pretty balanced stress shadings which are mostly somewhat lower than the maximum stresses of Z88P, however. The value of the order of integration INTORD in the header file Z88I3.TXT has no meaning but INTORD should be greater than 0.

4. The *von Mises* stresses as a mean value for each element are plotted. The stresses are computed in the Gauss points of the current element, added and then divided by the current number of Gauss points. This results in a mean value for the *von Mises stress* per element. The value of the order of integration INTORD in the header file Z88I3.TXT is important and INTORD must be greater than 0.
5. Plot of the displacements for X with color shading
6. Plot of the displacements for Y with color shading
7. Plot of the displacements for Z with color shading

For pos. 3. to 7. the color range may be edited in the header files Z88O.OGL (Windows) and Z88.FCD (UNIX).



UNIX: Hidden line plot of the BMW piston, mouse navigation turned on.

Drawing Node and Element Numbers:

Windows: *Labels > No Labels, Nodes, Elements,*
UNIX: Radio box *No Labels, Nodes, Elements*

Plot the element numbers or the node numbers or skip numbering. In contrary to Z88P you can define ranges *from-to*, e.g. plot the nodal numbers from 11 to 19 or plot the element 3, i.e. from 3 to 3. Z88O recalls your entries even if you change to *No Labels*. Keep in mind for UNIX and LINUX: press immediatelly the *Return* button after you've filled in a textfield to enter the value really. This is not a Z88O stupidy but the normal use of Motif textfields.
Please remember that you'll only get rendered all desired labels on the surfaces if you are in *Hidden line* mode. The other modes may cover some labels. And labels inside a structure are usually covered by the tetrahedron and hexahedron surfaces. Therefore, Z88O turnes the hidden line mode temporary off – you'll see the mesh without hidden lines just as in Z88P. As soon as you switch to *No Labels* gehen, the hidden line mode is re-activated.

Zooming:

Windows: *PRIOR* and *NEXT*
UNIX : Pushbuttons *Zoom+* and *Zoom-*
mouse navigation on: *left mouse button pressed*

Panning:

Windows: X: *CURSOR LEFT* and *CURSOR RIGHT*
Y: CURSOR UP and *CURSOR DOWN*
Z: HOME and *END*
UNIX : Pushbuttons *X+, X-, Y+, Y-, Z+, Z-*
mouse navigation on: *middle mouse button pressed*

Rotating:

Windows: *Faktors > Rotations 3D*
The rotations around X, Y and Z axises are defined with Factors > Rotations 3-D:ROTX, ROTY and ROTZ. Default values are 0.
With the F2 . . F7 keys the structure can be revolved in steps of 10 degrees.
UNIX : Pushbuttons *RX+, RX-, RY+, RY-, RZ+, RZ-*
Turn in steps of 10 degrees. Pushbutton *Rot 0* resets all rotations to 0.
mouse navigation on: *right mouse button pressed*

Enlarging Deflections:

Windows: *Factors > Deflections*
UNIX: Text fields *FUX, FUY* and *FUZ*

Enlarge the computed deflections with the factors FUX, FUY and FUZ. Default values are 100. Attention UNIX: As usual for UNIX, the changes only apply at a respective *Return*. However, you can type in all three fields without *Return* and then press the Pushbutton *Regen* (for Regenerate).

Some remarks on stresses:

Windows: *3D-Effects > Mises Stresses (corners)* or icon No. 9 from left
3D-Effects > Mises Stresses (elemente) or icon No. 10 from left

UNIX : Radiobutton *Stress C* (= corner nodes)
Radiobutton *Stress M* (= mean value per element)

If you did before a stress calculation with Z88D (this is possible and useful for all element types except for beams No.2, No.13 and cams No.5), then you may plot the *von Mises* stresses either in the corner nodes or as mean values per each element. And before running the stress processor Z88D you really had to calculate the displacements by running Z88F or the iteration solver. Thus, the sequence is:

1. Cholesky solver Z88F or iteration solver Z88I1 and Z88I2
2. stress processor Z88D
3. Z88O, if you want to plot stresses

von Mises stresses in corner nodes. In fact, the stresses are computed not really in the corner nodes which would lead to very wrong results especially for very tapered elements but in Gauss points laying in the near of the current corner nodes. Stresses are computed for just the same number of Gauss points like the number of corner points. Because often a node is linked to more than one element the stresses are computed to a mean value from the „corner node" stresses of all linked elements. This results in pretty balanced stress shadings which are mostly somewhat lower than the maximum stresses of Z88P, however. The value of the order of integration INTORD in the header file Z88I3.TXT has no meaning but INTORD should be greater than 0.

von Mises stresses as a mean value for each element. The stresses are computed in the Gauss points of the current element, added and then divided by the current number of Gauss points. This results in a mean value for the *von Mises stress* per element. The value of the order of integration INTORD in the header file Z88I3.TXT is important and INTORD must be greater than 0. Example: If you enter for INTORD the value 3 when calculating hexahedrons No.10 then the *von Mises* stresses are calculated in 3 x 3 x 3 = 27 Gauss point, added and then divided by 27.

The stress plot is done by color shading. You may modify the colors by editing the header files Z88O.OGL (Windows) or Z88.FCD (UNIX) for your own taste. Of course, for UNIX also the fonts – this requires some experiments depending on your operating system until you are (i hope so) satisfied. You need to make this modifications only once or leave it as it was.

Before that, run a stress calculation with Z88D. You did set in the header file Z88I3.TXT the stress flag *ISFLAG 1* and the integration order *INTORD > 0*. For exact stresses consult the output file Z88O3.TXT.

Caution: The operator is responsible for first running a stress calculation by Z88D before using this function. Run Z88F or the Iteration solver Z88I1 and Z88I2 and then Z88D before starting Z88O. Otherwise some old stress files Z88O8.TXT from earlier calculations are read in causing totally wrong results !!

Automatic Scaling:

Windows: *Autoscale > No Autoscale, Yes Autoscale*
UNIX: Pushbutton *AutoS*

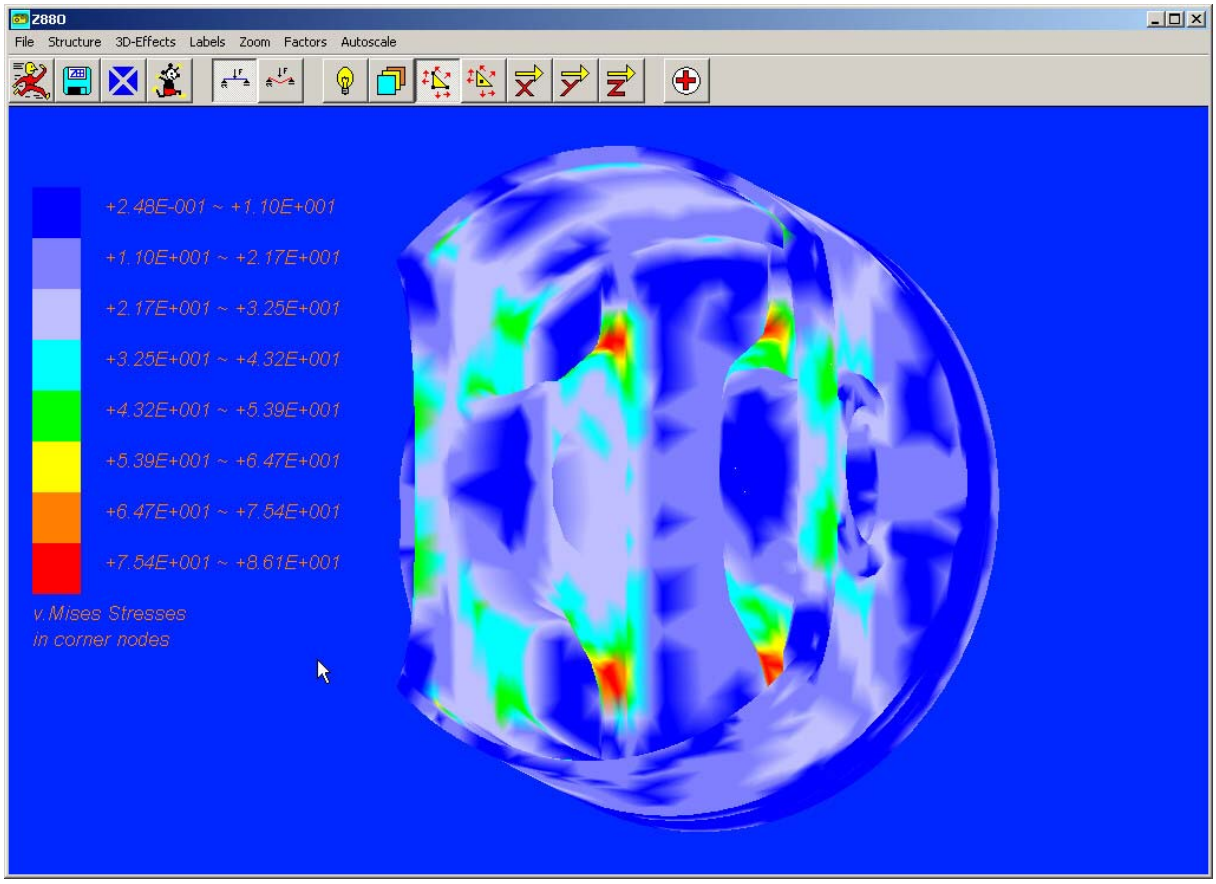
The Autoscale function takes care that structures will completely fit on the screen.

Autoscale activates automatically if a new structure is loaded by *File > Structure File*. Autoscale again is deactivated immediately and the hook points to *No Autoscale*. You can then scale properly with *Autoscale > Yes Autoscale*. However, Autoscale switches again immediately to *No Autoscale*. *Autoscale > Yes Autoscale* is a kind of pushbutton. The explanation above corresponds to UNIX as well.

Height Ratio:

Windows: file *Z88O.OGL*
UNIX: file *Z88.FCD*

The height ratio can be adjusted to the monitor customization. Therefore, the entry FYCOR exists in Z88O.OGL (Windows) or Z88.FCD (UNIX). Load a perfectly circular or perfectly square structure and modify FYCOR untill this structure is plotted perfectly circular or square on your monitor. Please keep in mind that FYCOR is loaded with the start of Z88O, so you must re-launch Z88O after a modification in the files. You need to make this modification only once.



Windows: plot of the von Mises stresses in the corner nodes of the BMW piston.

3 EDITING INPUT FILES

3.1 GENERAL INFORMATION

Z88 works with the following files:

(1) Input Files:

- *Z88I1.TXT (general structure data, coordinates, coincidence, material informations)*
- *Z88I2.TXT (boundary conditions, loads, constraints)*
- *Z88I3.TXT (parameters for stress processor Z88D)*
- *Z88I4.TXT (parameters for the iteration solver Z88I1/Z88I2)*
- *Z88NI.TXT (input file for the mesh generator)*

Produce these input files with your CAD program and the CAD converter Z88X or the COSMOS converter Z88G or with an editor (e.g. *Edit* or *Notepad* of Windows, *Vi*, *Emacs*, *Joe* at UNIX) or word processing program (e.g. *Wordpad* or *Word for Windows* at Windows). You can also use editors integrated in other programs, e.g. the program editors of compilers. If using a word processor systems keep in mind to edit pure ASCII texts without any hidden control characters ... every word processing program has such an option. Why not using your own editor (if you do not want or cannot work with a CAD program) ?

So how you can work with the editor/word processor you are familiar with and used to.

Details to the input files see sections 3.2 pp.

(2) Output Files:

- *Z88O0.TXT (processed input data for documentation)*
- *Z88O1.TXT (processed boundary conditions for documentation)*
- *Z88O2.TXT (calculated displacements)*
- *Z88O3.TXT (calculated stresses)*
- *Z88O4.TXT (calculated nodal forces)*

The file Z88O5.TXT is not a regular Z88 output file. It contains the coordinates of the stress points and the von Mises stresses and it is used for the plot program internally. It is a pure ASCII file, so that advanced users can use it for own routines, if necessary.
The plot program Z88P stores a HP-GL file, i.e. a plotter file which per default is called Z88O6.TXT. Other file names are possible.

(3) Binary Files:

These files are used internally and are not for editing. They serve the fast data interchange between Z88 modules.

- *Z88O1.BNY*
- *Z88O2.BNY (is only produced in Z88F New Mode for use of Z88F Old Mode)*
- *Z88O3.BNY*
- *Z88O4.BNY (only for internal communication of the iteration solver)*

Why work with files? Is that not old-fashioned and does "interactive" working not do a better job ? Z88 is designed as an open, transparent system according to the UNIX philosophy: Several, compact modules communicate via files together.

- **A maximum of memory is useable** for the FE data, because always only relatively small, compact programs are loaded into memory.

- Z88 is very flexible and adaptable through its open structure. **Any kind of preprocessing and postprocessing is possible without restrictions.** You can generate the input files by small, self-written preprograms (such a preprogram is the mesh generator Z88N) or leave the job of processing the output data to other programs: You can quite easily load Z88 output files into EXCEL and analyse there.

- Every FEA program can, and so does Z88, produce a huge amount of data junk from time to time. You are very often interested only in very specific results, e.g. of special nodes. The output files are simple ASCII files. You can edit and shorten them as you like and print only the **really interesting results**.

- Very often input files are produced **much faster** than by any interactive queries: Many input lines are similar to prior lines: Use the block operations of your editor for copying !

Downward compatibility:

Z88 files for Z88 V8.0, V8.0A, V8.0B, V9.0, V9.0B and V10.0 are okay for Z88 V11.0. It would be a good idea to add the plate flag to avoid confusion of the file checker Z88V.

Input file designed for Z88 versions *ealier than 8.0* cannot be processed by Z88 V8.0 because of changes according to coincidence and material informations for Z88I1.TXT and Z88NI.TXT. The input files Z88I2.TXT and Z88I3.TXT do have the same format as older versions. The former beam parameters file Z88I4.TXT is obsolete as beam parameters are now defined in Z88I1.TXT.

Rules for entering values:

There is no need for special rules or field divisions, only the usual C rules apply:

- *All values are to be separated by at least one blank*
- *Integer numbers may contain any point or exponents*
- *For floating point numbers no points need to be provided*
- *Numerical values which are 0 (zero), have to be entered explicitly.*

Integer numbers

Right	1	345	55555	0
Wrong	1.	345.	55555E+0	no entry

Floating point numbers (Z88 uses internally double precision floating point numbers *[Double]*)

Right	1.	345	5.5555E+10	0
Wrong	1,	345,	O (letter O)	no entry

Z88 input files may have comments in every line if all corresponding data has been filled out

before. Separate the last data and the comment at least by one blank. Lines in Z88 input files can include 250 bytes (really needed are noticeably less than 80). Blank lines and pure comment lines are not permitted.

Always check input files with Z88V before a Z88 run.

Z88V checks for formal correctness of the entry files. It can hardly recognize wrong or useless structures and boundary conditions. Examine when error messages or abnormal program stops of Z88:

- Are the files really pure text files, well in the ASCII format? Or have they been added unnoticed hidden control characters by your text processor ?

- Is the last line of an input file terminated by at least one *RETURN* ?

- Is MAXKOI in Z88.DYN large enough? If in doubt enter 1000000 or higher for MAXKOI.

- Is your structure statically determined or in any way statically overdefined (allowed !) ? Or is it statically indetermined, i. e. boundary conditions are missing which may cause serious trouble. Statically indetermined structures can appear easily for Beams No.2, Cams No.5 and Beams No.13 (take care of the rotation degrees of freedom).

- Is the coincidence list defined properly? Especially Hexahedrons No.10 are very sensitive to wrong numbering.

- Plot the initial structure with Z88O or Z88P. If you will not see some pretty good stuff, then the rest can hardly become better!

- Always do a rough calculation! Are the calculated deflections extremely high? Then check the boundary conditions quite carefully!

- And for the UNIX operating system: Are the file permissions properly set ? For the .LOG files, too ? Do a *chmod 777* !

- Z88 input files for UNIX and Windows have the same structure. You can without restriction load Z88-UNIX files into Windows and vice versa. But did you do the proper conversion ? Windows terminates lines by a CR/LF, but UNIX only by a LF ! Many UNIX systems feature the converters *unix2dos* and *dos2unix*.

3.2 GENERAL STRUCTURE DATA Z88I1.TXT

Mind the following formats:
[Long] = 4 bytes integer number
[Double] = 8 bytes floating point number, alternatively with or without point

1st input group, i. e. first line, contains:

Dimension of the structure (2 or 3)
Number of nodes of the FEA structure
Number of elements
Number of degrees of freedom
Number of material information lines
Coordinate flag KFLAG (0 or 1)
Beam flag IBFLAG (0 or 1)
Plate flag IPFLAG (0 or 1)

Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

Explanation KFLAG:
At input of 0 the coordinates are expected cartesian while at input of 1 polar or cylindrical coordinates are expected. The latter are then converted into cartesian coordinates and thereupon stored in this form in Z88O0.TXT. Caution: The axially symmetric elements No.6, 8 and 12 positively expect cylindrical coordinates, set KFLAG to 0 here!

Explanation IBFLAG:
If Beams No.2 or Beams No.13 appear in the structure, then set beam flag IBFLAG to 1, otherwise it must be 0.

Example: A three-dimensional structure of Hexahedrons No.10 and Beams No.2 is supposed to have 10 elements. The coordinates are entered in cartesian coordinates, 3 material info lines, 270 degrees of freedom and 45 nodes. Thus *: 3 45 10 270 3 0 1 0*

Explanation IPFLAG:
If Plates No.18, No.19 or No.20 appear in the structure, then set plate flag IPFLAG to 1, otherwise it must be 0.

Example: A two-dimensional structure of Plates No.20 is supposed to have 100 elements. The coordinates are entered in cylindrical coordinates, 2 material info lines, 540 degrees of freedom and 180 nodes. Thus *: 2 180 100 540 2 1 0 1*

Caution: This Z88 release allows only beams <u>or</u> plates in a structure, not both in the same structure!

2nd input group, starting with line 2, contains:

Coordinates, one line per node.

Node number, strictly ascending [Long]
Number of the degrees of freedom for this node [Long]
X-coordinate or, if KFLAG is 1, R- coordinate [Double]
Y-coordinate or, if KFLAG is 1, PHI-coordinate [Double]
Z-coordinate or, if KFLAG is 1, Z-coordinate [Double]
The Z coordinate can be dropped at 2-dimensionalen structures. Enter angles PHI in radian.

Write all numbers into a line, separate at least by one blank respectively.

Example 1: The node no.156 has 2 degrees of freedom and the coordinates X = 45.3 and Y = 89.7 . Thus *: 156 2 45.3 89.7*

Example 2: The node no.68 is supposed to have 6 degrees of freedom (a Beam No.2 is attached) and cylindrical coordinates R = 100. , PHI = 0.7854 (corresponds to 45 °), Z = 56.87. Thus
68 6 100. 0.7854 56.87

3rd input group, starting after last node, contains:

Coincidence, two lines for every finite element

1st line:
Element number, strictly ascending
Element type (1 to 20)

Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

2nd line: Depending on element type
1st node number for coincidence
2nd node number for coincidence
.....
20th node number for coincidence

Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

Example: An Isoparametric Serendipity Plane Stress Element No.7 has element number 23. The coincidence has the global nodes 14, 8, 17, 20, 38, 51, 55, 34 (locally these are the nodes 1-2-3-4-5-6-7-8, see chapter 4.7) . Thus resulting in two lines:
23 7
14 8 17 20 38 51 55 34

4th input group, starting after last element, contains:
Material information, one line for each material information.

This material information line starts with element no. inclusively [Long]
This material information line ends with element no. inclusively [Long]
Youngs's Modulus [Double]
Poisson's Ratio [Double]
Integration order (0, 1, 2, 3, 4, 5, 7 or 13) [Long]
Cross section value QPARA [Double]

... And if beams (but not plates !) are defined in addition:
Second moment of inertia yy (bending around yy axis)
Max. distance from neutral axis yy
Second moment of inertia zz (bending around zz axis)
Max. distance from neutral axis zz
Second moment of area (torsion)
Second modulus (torsion)

... And if plates (but not beams !) are defined in addition:
area load

Write all numbers into a line, separate at least by one blank respectively.
Explanation cross section value QPARA:
QPARA is element type-dependent, e.g. for hexahedrons QPARA is 0, for trusses QPARA is the cross-sectional area and for plane stress elements QPARA is the thickness. See chapter 4.

Example: The structure has 34 finite elements No.7. The thicknesses is supposed to vary: Elements 1 to 11 thickness 10 mm, elements 12 to 28 15 mm and elements 29 to 34 now 18 mm. Material steel. Integration order is supposed to be 2. Thus three material information lines:

```
1    1    11   206000   0.3   2   10.
2    12   28   206000   0.3   2   15.
3    29   34   206000   0.3   2   18.
```

3.3 MESH GENERATOR INPUT FILE Z88NI.TXT

The layout of Z88NI.TXT is very similar to the layout of Z88I1.TXT, the input file for the FE processor: Only the &- labeled data is required in addition. Cause: Z88NI.TXT can serve the plot program Z88P. Moreover, Z88NI.TXT can be copied to the name Z88I1.TXT and therefore be used to feed the FE processor with a very rough structure for very first checks and results. Mind the following formats:
[Long] = 4 bytes integer number
[Double] = 8 bytes floating point number, alternatively with or without point
[Character] = A letter

1st input group, i. e. first line, contains:

Dimension of the structure (2 or 3)
Number of nodes of the super structure
Number of super-elements
Number of degrees of freedom
Number of material information lines
Coordinate flag KFLAG (0 or 1)
Beam flag (must be 0 here !)
Plate flag (0 or 1)
& Trap radius flag NIFLAG (0 or 1)
Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

Explanation KFLAG:
At input of 0 the coordinates are expected cartesian while at input of 1 polar or cylindrical coordinates are expected. The latter are then converted into cartesian coordinates and thereupon stored in this form in Z88I1.TXT. Caution: The axially symmetric elements No.8 and 12 positively expect cylindrical coordinates, set KFLAG to 0 here !

Explanation IPFLAG:
If Plates No.20 appear in the structure, then set plate flag IPFLAG to 1, otherwise it must be 0.

Explanation NIFLAG:
In order to identify already defined nodes the mesh generator needs a trap radius. The defaults are 0.01 for for EPSX, EPSY and EPSZ if NIFLAG is 0. These values can be modified at extremely small or large structures. To initiate this change, set NIFLAG to 1. The new trap radiuses of EPSX, EPSY and EPSZ are then defined in Z88NI.TXT as the 6th input group.

Example: Super-structure 2-dimensional with 37 nodes, 7 super elements, 74 degrees of freedom, one material information line. Cartesian coordinates, no beams (anyway forbidden in the mesh generator file), trap radius default value. Thus
2 37 7 74 1 0 0 0 0

2nd input group, starting in line 2, contains:
Coordinates, one line per node.

Node number, strictly increasing [Long]
Number of the degrees of freedom for this node [Long]
X-coordinate or, if KFLAG is 1, R- coordinate [Double]
Y-coordinate or, if KFLAG is 1, PHI-coordinate [Double]
Z-coordinate or, if KFLAG is 1, Z-coordinate [Double]
The Z coordinate can be skipped at 2-dimensional structures. Enter angles PHI in radian.
Write all numbers into a line, separate at least by one blank respectively.

Example: The node no.8 has 3 degrees of freedom and the coordinates X = 112.45, Y = 0. , Z = 56.75. Thus*: 8 3 112.45 0. 56.75*

3rd input group, starting after last node, contains:
Coincidence, two lines for every finite element

1st line:
Element number, strictly ascending
Super-element type (7,8,10,11,12,20) [Long]

Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

2nd line: Depending on element type
1st node number for coincidence
2nd node number for coincidence
.....
20th node number for coincidence

Write all numbers into a line, separate at least by one blank respectively. All numbers here of the type [Long].

Example: An Isoparametric Serendipity Plane Stress Element No.7 has element number 23. The coincidence has the global nodes 14, 8, 17, 20, 38, 51, 55, 34 (locally these are the nodes 1-2-3-4-5-6-7-8, see chapter 4.7) . Thus resulting in two lines:
23 7
14 8 17 20 38 51 55 34

4th input group, starting after last element, contains:

Material information, one line for each material information.
This material information line starts with super-element no. inclusively [Long]
This material information line ends with super-element no. inclusively [Long]
Youngs's Modulus [Double]
Poisson's Ratio [Double]
Integration order (1, 2, 3 or 4) [Long]
Cross section value QPARA [Double]
... And if plates are defined in addition:
area load

Write all numbers into a line, separate at least by one blank respectively. Beams and cams are

forbidden in Z88NI.TXT.

Explanation cross section value QPARA:

QPARA is element type-dependent, e.g. for hexahedrons 0, for trusses the cross-sectional area, and for plane stress elements the thickness. Here are the mesh generator-suitable elements:

- Element No.1: Isoparametric Hexahedrons 8 nodes
- Element No.7: Isoparametric Serendipity Plane Stress Element 8 nodes
- Element No.8: Isoparametric SerendipityTorus 8 nodes
- Element No.10: Isoparametric Serendipity Hexahedron 20 nodes
- Element No.11: Isoparametric Serendipity Plane Stress Element 12 nodes
- Element No.12: Isoparametric Serendipity Torus 12 nodes
- Element No.20: Isoparametric Serendipity Plate 8 nodes

Example: The structure has 34 super elements No.7. The thicknesses are supposed to vary: Elements 1 to 11 thickness 10 mm, elements 12 to 28 15 mm and elements 29 to 34 now 18 mm. Material steel. Integration order shall be 2. Thus three material information lines:

1 1 11 206000 0.3 2 10.
2 12 28 206000 0.3 2 15.
3 29 34 206000 0.3 2 18.

& 5th input group, starting after last material information line, contains:

The descriptive details for the mesh generation process. 2 lines for every super element.

1st line:
Super element no. [Long]
Finite element type(types 1,7,8,10,19,20)to be generated [Long]

2nd line:
Number of finite elements in local x direction [Long]
Type of subdivision of CMODE x [Character]
Number of finite elements in local y direction [Long]
Type of the subdivision CMODE y [Character]
Number of finite elements in local z direction [Long]
Type of the subdivision of CMODE z [Character]

The two values for Z are skipped at 2-dimensional structures.

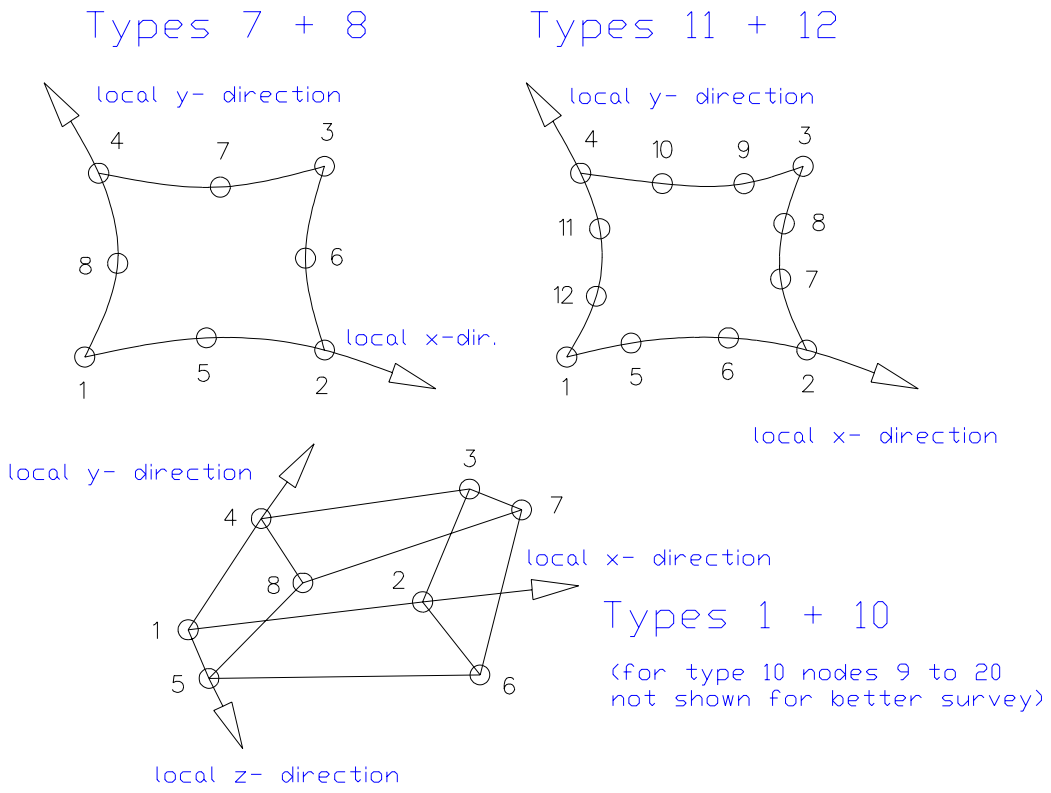
Explanations: CMODE can accept the following values:

- "E": Subdivision equidistant, "e" is also permitted
- "L": Subdivision increasing geometrically in local coordinate direction
- "l": Subdivision decreasing geometrically in local coordinate direction

The local x-, y and z axises are defined as follows:

- Local x axis points in direction of local nodes 1 and 2
- Local y axis points in direction of local nodes 1 and 4
- Local z axis points in direction of local nodes 1 and 5

See following sketch:



Example: Subdivide an Isoparametric Serendipity Plane Stress Element with 12 nodes (Element No.11) into finite elements of type Isoparametric Serendipity Plane Stress Element with 8 nodes (Element No.7). Subdivide in local x direction three times equidistantly and subdivide 5 times increasing geometrically in local y direction. The super element is supposed to have the number 31. Thus :
31 11
7 3 E 5 L (e or E for equidistant are equivalent)

& 6th input group, optionally after the end of input group 5 :

Input group 6 is required if NIFLAG was set to 1, i. e. the trap radiuses is upposed to be modified. 1 line :

Trap radius in global X direction EPSX [Double]
Trap radius in global Y direction EPSY [Double]
Trap radius in global Z directionEPSZ [Double]
Skip the Z detail for 2-dimensionalen structures.

Example: The trap radiuses shall be set to 0.0000003 for X, Y and Z respectively. Thus :
0.0000003 0.0000003 0.0000003

This is effective only if NIFLAG was set to 1 in the first input group!

3.4 BOUNDARY CONDITIONS Z88I2.TXT

Mind the following formats:
[Long] = 4 bytes integer number
[Double] = 8 bytes floating point number, alternatively with or without point

1st input group, i. e. first line, contains:

Number of the boundary conditions: loads and constraints [Long]

2nd input group, starting in line 2, contains:

Boundary conditions and loads. For every boundary condition and for every load respectively one line.

node number with boundary condition: load or constraint [Long]
Respective degree of freedom (1,2,3,4,5,6) [Long]
Header flag: 1 = force [Long] or 2 = displacement [Long]
Value of the load or displacement [Double]

Example: The node 1 shall be fixed respectively at his 3 degrees of freedom: support. Node 3 gets a load of -1,648 N in Y direction (i.e. DOF 2), the degrees of freedom 2 and 3 is supposed to be fixed for the node 5. Resulting in 6 boundary conditions. Thus :

```
6
1    1    2    0
1    2    2    0
1    3    2    0
3    2    1    -1648
5    2    2    0
5    3    2    0
```

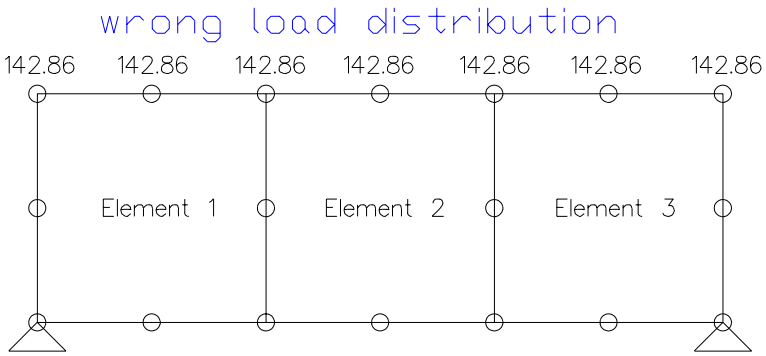
For edge loads, area loads and surface loads pay attention to:

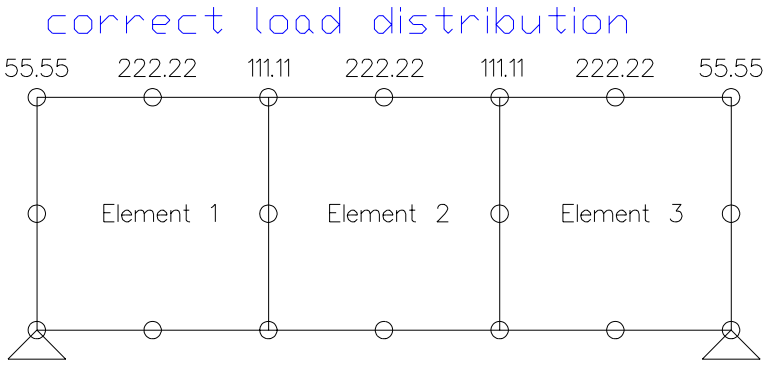
Area loads for plates No.18, No.19 and No.20 may entered directly in the material information line for the appropriate elements (file Z88I1.TXT). Only forces should entered here into Z88I2.TXT.

For the elements with linear shape function, e.g. Hexahedrons No.1 and Torus No.6, edge loads, area loads and surface loads are distributed to the elements simply and straightly onto the respective nodes.

However, for elements with higher shape functions, i. e. square (Plane Stress No.3, No.7, Torus No.8, Hexahedron No.10) or cubic (Plane Stress No.11 and Torus No.12) edge and surface loads have to be put onto the elements according to fixed rules which are not always physically obvious. Really funny, some load components can have negative values. Though these facts are not obvious, nethertheless they lead to correct results which is not the case for intuitive distribution of loads to the respective nodes.

An example shall clarify the facts:

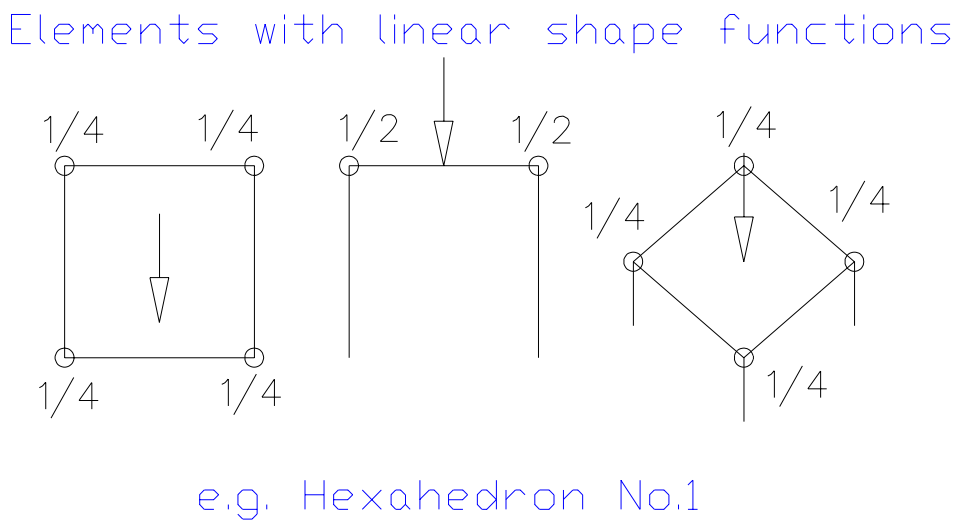




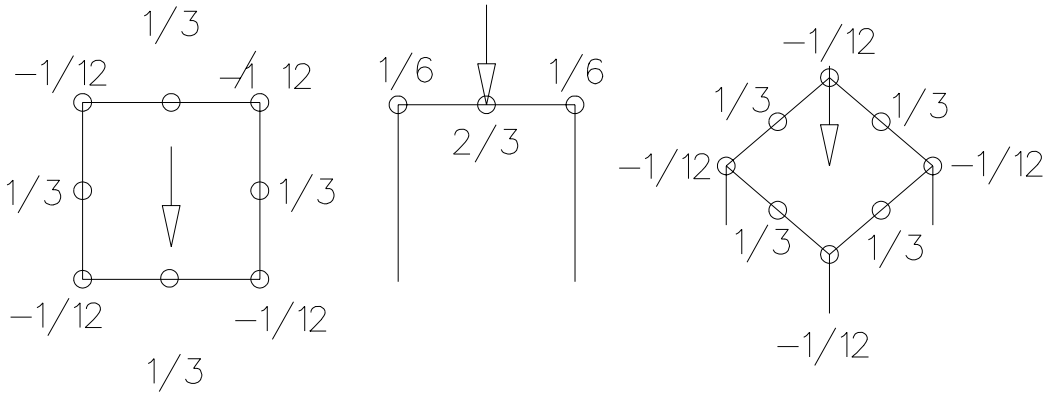
A FE structure consists of three plane stress elements No.7 with the load of 1,000 N distributed on the upper edge in Y direction. Above incorrect, below correct load sharing:

Incorrect: 1,000N/7=142.86 N per node. Not correct for elements with square shape function.

Correct: 2 * 1/6 + 2 * (1/6+1/6) + 3 * 2/3 = 18/6 = 3, corresponds to 1,000 N
"1/6 points" = 1,000/18*1 = 55.55
"2/6 points" = 1,000/18*2 = 111.11
"2/3 points" = 1,000/18*4 = 222.22
Control: 2*55.55 + 2*111.11 +3*222.22 = 1,000 N, o.k. Here's why:

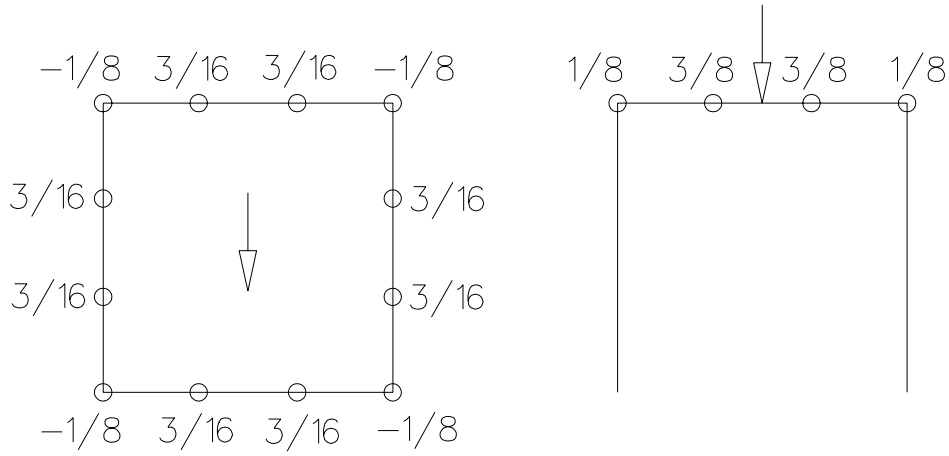


Elements with quadratic shape functions



e.g. plane stress element No.3 and No.7
Torus No.8 and Hexahedron No.10

Elements with cubic shape functions



e.g. plane stress element No.11 and Torus No.12

3.5 STRESS PARAMETER FILE Z88I3.TXT

Mind the following format:
[Long] = 4 bytes Integer number

File only consists of only one line:

1st value: For isoparametric elements No. 1, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19, 20:
Value of the integration order INTORD [Long]

Valid is:

0 = Calculation of stresses into the corner nodes, *von Mises* stress calculation not possible.

For isoparametric elements No.1, 7, 8, 10, 11, 12, 19, 20:

1, 2, 3 or 4 (i.e. N*N) = Calculation of stresses into the Gauss points. *Von Mises* stress calculation is possible. A good value is 3 (= 3*3 Gauss points). For element type No.1 and No.20 a value of 2 could be fine. For type No.19 a value of 4 (= 4*4 Gauss points) is recommended.

For isoparametric elements No.14, 15, 18:

3, 7 or 13 (i.e. N) = Calculation of stresses into the Gauss points. *Von Mises* stress calculation is possible. A good value is 7 (= 7 Gauss points). For type No.18 a value of 3, i.e. 3 Gauss points, could be fine.

For isoparametric elements No.16, 17:

1, 4 or 5 (i.e. N) = Calculation of stresses into the Gauss points. Von Mises stress calculation is possible. A good value is 5 (= 5 Gauss points). For element type No.17 a value of 1 could be fine.

This 1st value has no meaning for element types No.2, 3, 4, 5, 6, 9 and 13. However, please enter a value of 1 to satisfy the filechecker Z88V.

2nd value: *For the plane stress elements No.3, 7, 11 and 14: KFLAG* [Long]

0 = standard stress calculation
1 = additional calculation of the radial and tangential stresses

3rd value: *Choice of the reduced stress hypothesis: ISFLAG* [Long]

0 = no calculation of the reduced stresses
1 = *von Mises* stresses

Example 1: The stress processor Z88D is supposed to calculate for a structure of Plane Stress Elements No.7 the stresses for every finite element into 3*3 Gauss points: INTORD = 3. In addition to this calculation of standard-stresses a calculation of radial and tangential stresses

is supposed to be run, KFLAG = 1. Furthermore compute *von Mises* stresses: ISFLAG = 1. Thus : *3 1 1*

Example 2: The stress processor Z88D is supposed to compute only the stresses of the corner nodes for every finite element No.7. Only standard-stress calculation, thus KFLAG = 0. Do not compute *von Mises* stresses, thus ISFLAG = 0. Thus : *0 0 0*

3.6 PARAMETER FILE Z88I4.TXT FOR THE ITERATION SOLVER PART 2: Z88I2

Mind the following formats:
[Long] = 4 bytes integer number
[Double] = 8 bytes floating point number, alternatively with or without point

File only consists of only one line:

1st entry: *Number of iterations MAXIT [Long]*. When Z88I2 reaches this value, the solver is halted in any case. The results reached to this point are printed into Z88O2.TXT, however. This is the first halt criterion. Enter a value not too small e.g. 10000.

2nd entry: *Limit EPS [Double]*. This value is compared to a norm of the residual vector. When reaching this limit, the solution may have a good precision. This is the second halt criterion. Enter a relatively small value, e.g. 0.00001 or 0.0000001. This are quite proper and tested values. *Note that there is no absolute truth in this field! Which ever norm of the residual vector is compared against the linit EPS - you can never be sure that all elements of the solution vector are precise.* The choice of EPS has heavy influence on the iteration count and, thus, on the computing speed. Remember this when comparing Z88 to the big, commercial solvers (you don't really know which halt criterions these folks have programmed). The limits you may adjust in the commercial solvers may have nothing to do with EPS of Z88. However, many Z88- tests proved that the deflections of different nodes compared quite well to those from the commercial solvers if EPS was between 0.00001 and 0.0000001 with similar elapsed time. And pay attention to the fact, that you'll never know which solver delivers the best results when computing a large FEA structure!

3rd entry : *Convergence acceleration parameter RP [Double]*. Depends on your choice of preconditioner (the solver works with *Conjugate Gradients*, however).

- in case of SOR: Relaxation factor *Omega* (from 0 to 2, good values may vary from 0.8 to 1.2).
- in case of SIC: Shift factor *Alpha* (from 0 to 1, good values may vary from 0.0001 to 0.1). For further information consult the special literature)

I recommend SORCG (Conjugate Gradients with SOR preconditioning) as the default solver, because this solver needs only about 2/3 of memory of the second solver SICCG (shifted incomplete Cholesky decomposition, Conjugate Gradients). Which value may you choose for *RP* (reads here *Omega*) ? Good question ! Try *RP* with *1*, this won't result in too bad results and vary RP for further runs with this FEA structure.

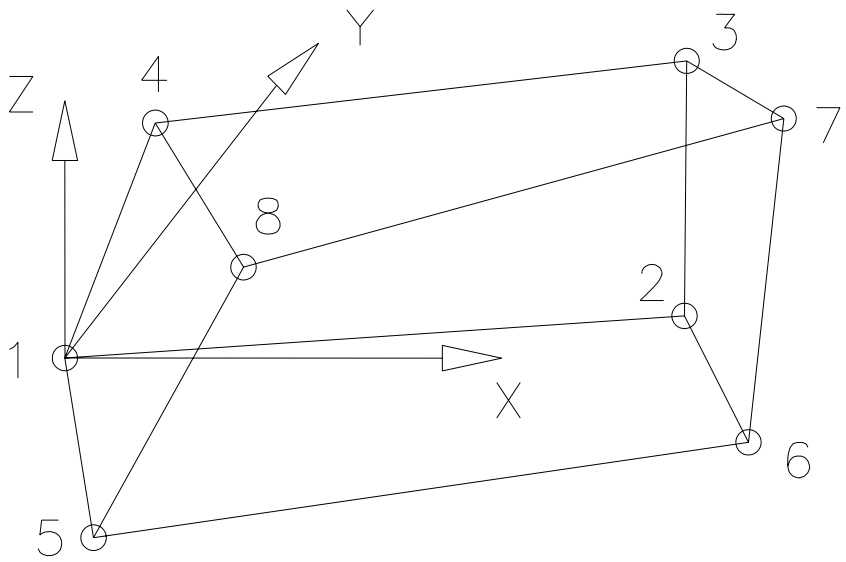
Example: *You want to stop after 5000 iterations, you choose a limit of 0.00001and the convergence acceleration parameter will be 1 for use with SORCG solver.*
> Thus: 5000 0.00001 1.

4 DESCRIPTION OF THE FINITE ELEMENTS

4.1 HEXAHEDRON NO.1 WITH 8 NODES

The hexahedron element calculates deflections and stresses in space. It is a transformed element, therefore it can have a wedging form or another oblique-angled form. The transformation is isoparametric. The integration is carried out numerically in all three axises according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 2 is mostly sufficient. Hexahedron No.1 is also well usable as a thick plate element, if the plate's thickness is not too small against the other dimensions. The element causes high computing load and needs a lot of memory, because the element stiffness matrix has the order 24*24.

Hexahedrons No.1 can be generated by the net generator Z88N from super elements Hexahedrons No.10, but Hexahedron No.1 cannot be used as a super element.



Input:

CAD (see chapter 2.7.2):

Upper plane: 1 - 2 - 3 - 4 - 1, quit LINE function
Lower plane: 5 - 6 -7 - 8 - 5, quit LINE function
1 - 5, quit LINE function
2 - 6, quit LINE function
3 - 7, quit LINE function
4 - 8, quit LINE function

Z88I1.TXT

> KFLAG for cartesian (0) or cylindrical coordinates (1)
> 3 degrees of freedom for each node
> Element type is 1
> 8 nodes per element
> Cross-section parameter QPARA is 0 or any other value, has no influence
> Integration order INTORD for each mat info line. 2 is usually good.

Z88I3.TXT

> *Integration order INTORD* for stress calculation:
Can be different from INTORD in Z88I1.TXT.
0 = Calculation of stresses in the corner nodes
1,2,3,4 = Calculation of stresses in the Gauss points

> any *KFLAG*, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses in the Gauss points (INTORD not 0 !)

Results:

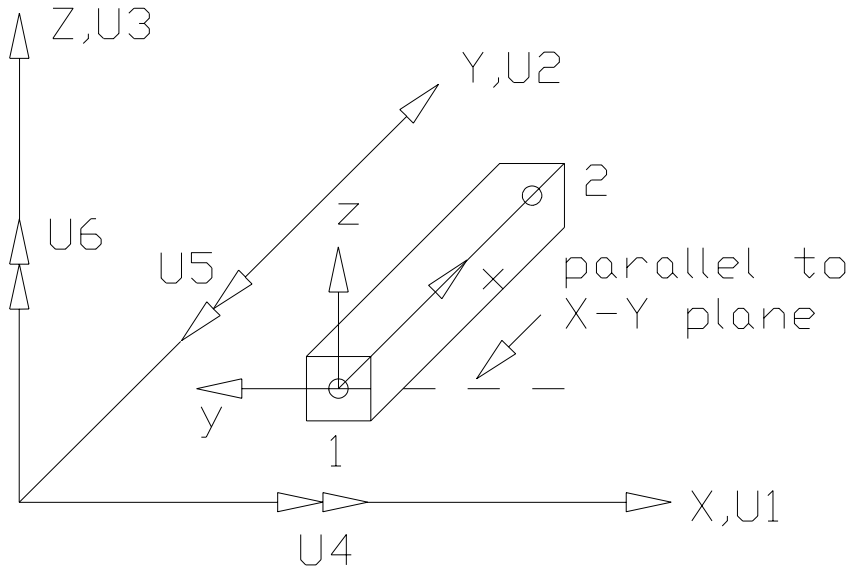
Displacements in X, Y and Z

Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises stresses.

Nodal forces in X, Y and Z for each element and each node.

4.2 BEAM NO.2 WITH 2 NODES IN SPACE

Beam element with any symmetric profile (no slanting bend) with the restriction that the local y-y axis must be parallel to the global X-Y coordinate system. The profile values are provided in Z88I1.TXT. Thus, you can use any symmetric profile in contrast to other FEA programs which sometimes incorporate a variety of different special beam and profile subroutines without matching all symmetric profiles as necessary . The element matches exactly Bernoulli's bend theory and Hooke's law. It uses no approximate solution as for the continuum elements.



Input:

CAD (see chapter 2.7.2): *Line from node 1 to node 2*

Z88I1.TXT

> KFLAG for cartesian (0) or cylindrical coordinates (1)
> Set beam flag IBFLAG to 1
> 6 degrees of freedom in a node (Attention: DOF5 (not right hand rule), see below)
> Element type is 2
> 2 nodes per element

At the material information lines:
> Integration order INTORD is arbitrary (1..4), has no influence
> Cross-sectional area QPARA
> Second moment of inertia RIYY (bending around y-y axis)
> Max. distance EYY from neutral axis y-y
> Second moment of inertia RIZZ (bending around z-z axis)
> Max. distance EZZ from neutral axis z-z
> Second moment of area (torsion) RIT
> Second modulus (torsion) WT
Z88I3.TXT

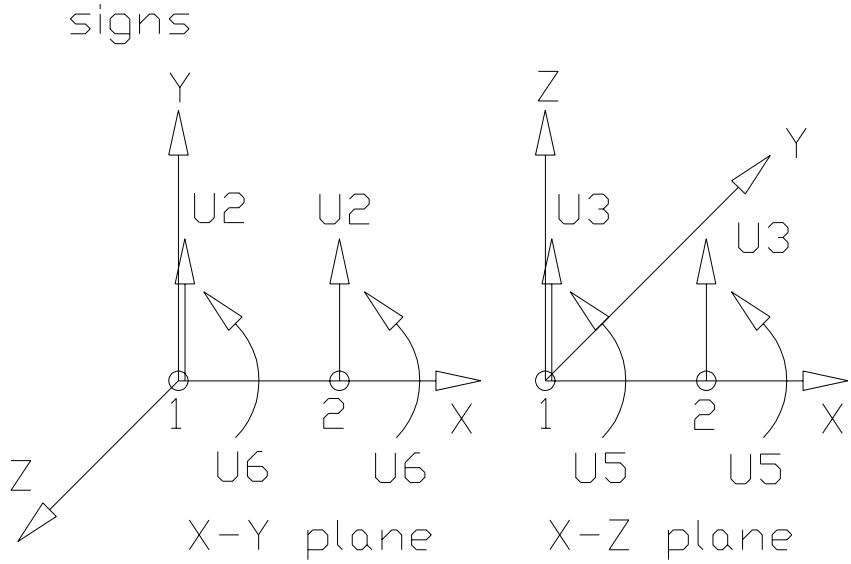
Beams No.2 have no influence. However, Z88I3.TXT must exist (with any content).

Results:

Deflections in X, Y and Z and rotations around X, Y and Z. Attention DOF5 (not right hand rule), see below

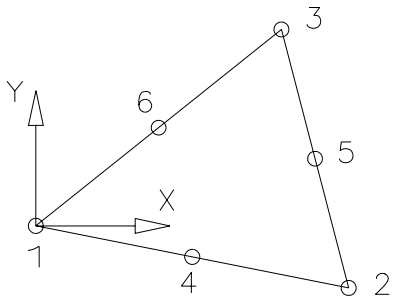
Stresses: SIGXX, TAUXX: Direct stress, shear stress, SIGZZ1, SIGZZ2: Bending stress around z-z for node 1 and node 2, SIGYY1 SIGYY2: Bending stress around y-y for node 1 and node 2

Nodal forces in X, Y and Z and nodal moments around X, Y and Z for each element and each node.



4.3 PLANE STRESS TRIANGLE NO.3 WITH 6 NODES

This is a simple, triangular plane stress element with complete square shape functions. Pay attention to edge loads, cf. chapter 3.4.



Input:

CAD (see chapter 2.7.2): *1-4-2-5-3-6-1*

Z88I1.TXT

> *KFLAG for cartesian (0) or polar coordinates (1)*
> *2 degrees of freedom for each node*
> *Element type is 3*
> *6 nodes per element*
> *Cross-section parameter QPARA is the element thickness*

Z88I3.TXT

> *Integration order INTORD:* any order, has no influence

> *KFLAG* = 0: Calculation of SIGXX, SIGYY and TAUXY
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses

Results:

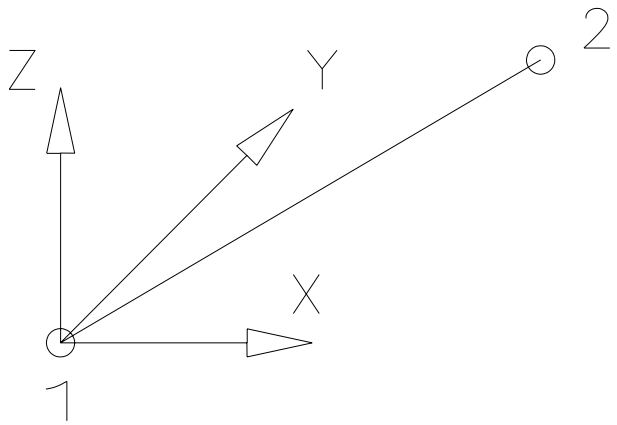
Displacements into X and Y

Stresses: The stresses are calculated in the element's center of gravity. The coordinates of the center of gravity are thus printed. For KFLAG = 1 the radial stresses SIGRR, the tangential stresses SIGTT and the accompanying shear stresses SIGRT are computed additionally (makes only sense if a rotational-symmetric structure is available). For easier orientation the respective radiuses and angles of the center of gravity are printed. Optional von Mises stresses in the center of gravity.

Nodal forces in X and Y for each element and each node.

4.4 TRUSS NO.4 IN SPACE

The truss element No.4 can take any location in space. It is part of the simplest elements in Z88 and is calculated extremely fast. The truss elements matches Hooke's law exactly .



Input:

CAD (see chapter 2.7.2): *Line from node 1 to node 2*

Z88I1.TXT

> KFLAG for cartesian (0) or cylindrical coordinates (1)
> 3 degrees of freedom for each node
> Element type is 4
> 2 nodes per element
> Cross-section parameter QPARA is the cross-sectional area of the truss

Z88I3.TXT

Trusses No.4 have no influence. However, Z88I3.TXT must exist (with any content).

Results:

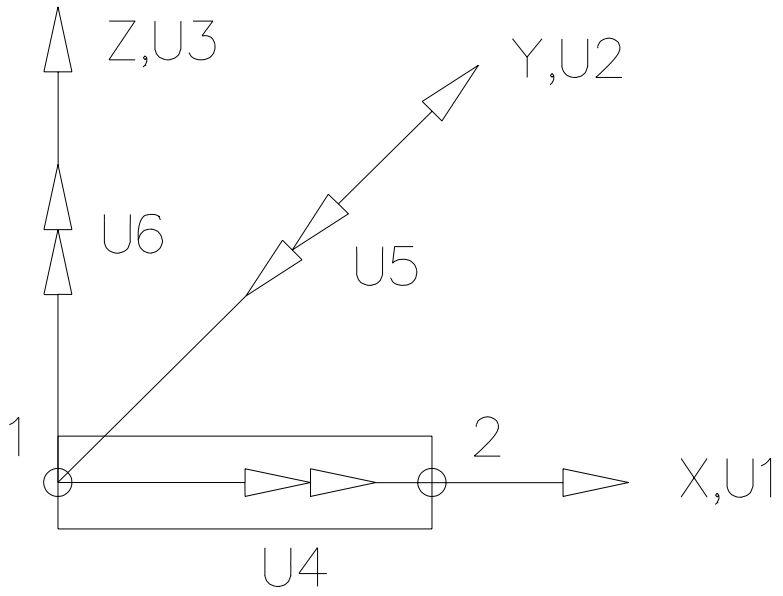
Displacements in X, Y and Z

Stresses: Normal stresses

Nodal forces in X, Y and Z for each element and each node.

4.5 CAM ELEMENT NO.5 WITH 2 NODES

The cam element is a simplification of the general beam element No.2: It has always a circular cross-cut. The element lies concentrically to the X axis, consequently local and global coordinates have the same direction. Inputs and calculations are simplified strongly through this. Like with the beam element the results are exact according to Bernoulli's bend theory and Hooke's law, and not approximate solutions like with the continuum elements.



Input:

CAD (see chapter 2.7.2): *Line from node 1 to node 2*

Z88I1.TXT

> Set KFLAG on 0 for cartesian coordinates
> 6 degrees of freedom in a node (Attention DOF5 (not right hand rule), see below)
> Element type is 5
> 2 nodes per element
> Cross-section parameter QPARA is the diameter of the cam

Z88I3.TXT

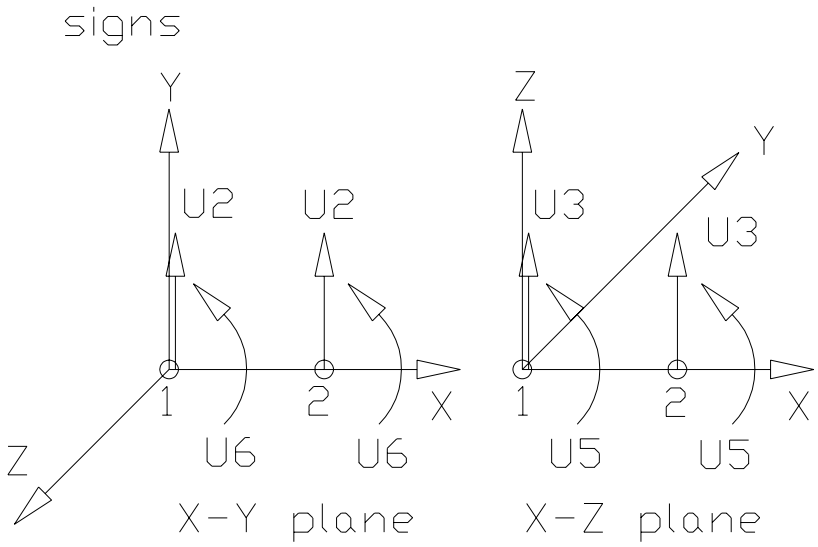
Cams No.2 have no influence. However, Z88I3.TXT must exist (with any content).

Results:

Deflections in X, Y and Z and rotations around X, Y and Z. Attention DOF5 (not right hand rule), see below

Stresses: SIGXX, TAUXX: Direct stress, shear stress, SIGXY1, SIGXY2: Bending stress in X-Y plane for node 1 and node 2, SIGXZ1 SIGXZ2: Bending stress in X-Z plane for node 1 and node 2

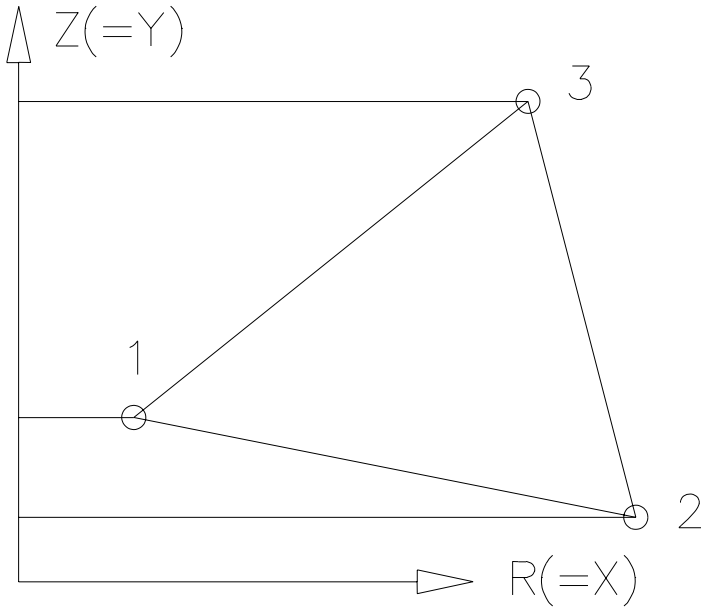
Nodal forces in X, Y and Z and **nodal moments** around X, Y and Z for each element and each node.



4.6 TORUS NO.6 WITH 3 NODES

This element is implemented only for historical reasons and possible data exchange to other FEA systems. Much better: Torus No.8 or Torus No.11.

This is a simple, triangular torus element with linear shape functions for rotational-symmetric structures. The displacement results for this very simple element are quite useable, but the stress calculation results are inaccurate. The stresses are calculated in the corner nodes internally and then distributed as average value in the center of gravity. However, the use of the torus elements No.8 or No.12. is highly recommended especially for accurate stress calculations.



Input:

CAD (see chapter 2.7.2): *1-2-3-1*

Z88I1.TXT

> In principle cylindrical coordinates are expected: KFLAG must be 0 !
R coordinate (= X), always positive
Z coordinate (= Y), always positive
> 2 degrees of freedom for each node, DOF R and Z (= X and Y)).
> Element type is 6
> 3 nodes per element
> Cross-section parameter QPARA is 0 or any value, no influence

Z88I3.TXT

> *INTORD* , any, has no influence
> *KFLAG* , any, has no influence
> *Reduced stress flag ISFLAG:*
0 = no reduced stress calculation
1 = von Mises stresses, plotted in the center of gravity

Results:

Displacements in R and Z (= X and Y)

Stresses: The stresss are internally computed in the corner nodes, but plotted in the center of gravity.

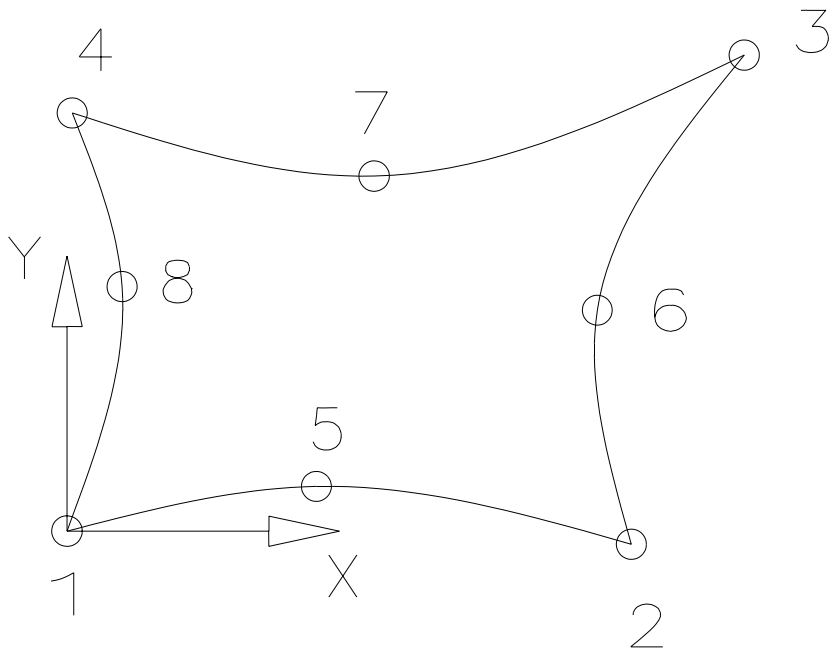
It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), SIGTE = stress in peripherical direction = tangential stress. Optional von Mises stresses.

Nodal forces for each element and each node.

4.7 PLANE STRESS ELEMENT NO.7 WITH 8 NODES

This is a curvilinear Serendipity plane stress element with square shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 is mostly sufficient. This element calculates both displacements and stresses very exactly. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Pay attention to edge loads, cf. chapter 3.4.
Plane Stress Elements No.7 can be generated by the net generator Z88N from super elements Plane Stress Elements No.7 or No.11. Thus, the Plane Stress Element No.7 is well suited as super element.

Plane Stress Element No.7 is recommended for all sort of plane stress computation. This element is well-balanced in respect to the precision of displacement and stress calculation as well as to its needs for memory and computing power.



Input:

CAD (see chapter 2.7.2): *1-5-2-6-3-7-4-8-1*

Z88I1.TXT

> KFLAG for cartesian (0) or polar coordinates (1)
> 2 degrees of freedom for each node
> Element type is 7
> 8 nodes per element
> Cross-section parameter QPARA is the element thickness
> Integration order INTORD per each mat info line. 3 is usually good.

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted
0 = Calculation of the stresses in the corner nodes
1,2,3,4 = Calculation of the stresses in the Gauss points

> *KFLAG* = 0: Calculation of SIGXX, SIGYY and TAUXY
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements ino X and Y.

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. For KFLAG = 1 the radial stresses SIGRR, the tangential stresses SIGTT and the accompanying shear stresses SIGRT are computed additionally (makes only sense if a rotational-symmetric structure is available). For easier orientation the respective radiuses and angles of the nodes/points are printed. Optional von Mises stresses

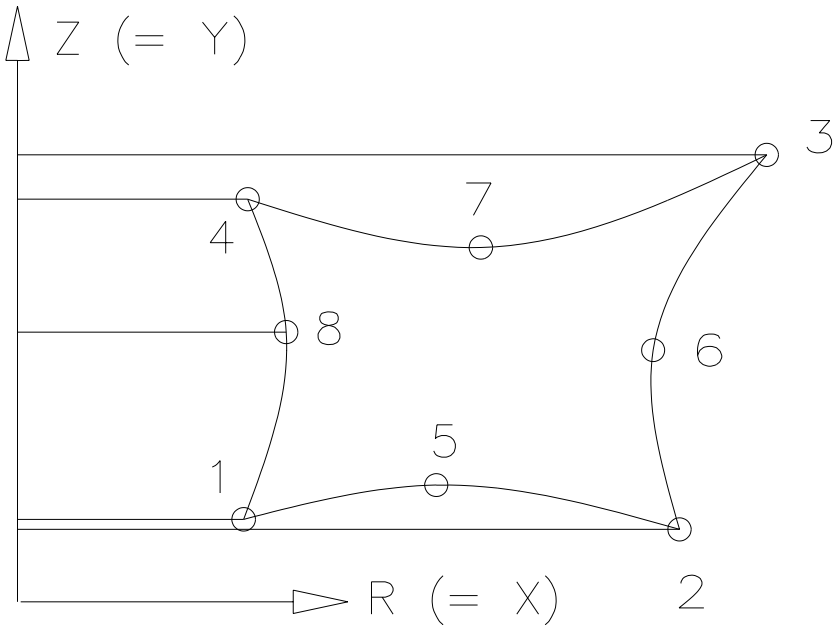
Nodal forces in X and Y for each element and each node.

4.8 TORUS NO.8 WITH 8 NODES

This is a curvilinear Serendipity torus element with square shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 is mostly sufficient. This element calculates both displacements and stresses very exactly. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Pay attention to edge loads, cf. chapter 3.4.

Torus elements No.8 can be generated by the net generator Z88N from the super elements torus elements No.8 or No.12. Thus, Torus No.8 is well suited as super element.

Torus element No.8 is recommended for all sort of axialsymmetric computation. This element is well-balanced in respect to the precision of displacement and stress calculation as well as to its needs for memory and computing power.



Input:

CAD (see chapter 2.7.2): *1-5-2-6-3-7-4-8-1*

Z88I1.TXT

> In principle cylindrical coordinates are expected: KFLAG must be 0 !
R coordinate (= X), always positive
Z coordinate (= Y), always positive
> 2 degrees of freedom for each node, DOF R and Z (= X and Y).
> Element type is 8
> 8 nodes per element
> Cross-section parameter QPARA is 0 or any value, no influence
> Integration order per each mat info line. 3 is usually good.

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted
0 = Calculation of the stresses in the corner nodes
1,2,3,4 = Calculation of the stresses in the Gauss points

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

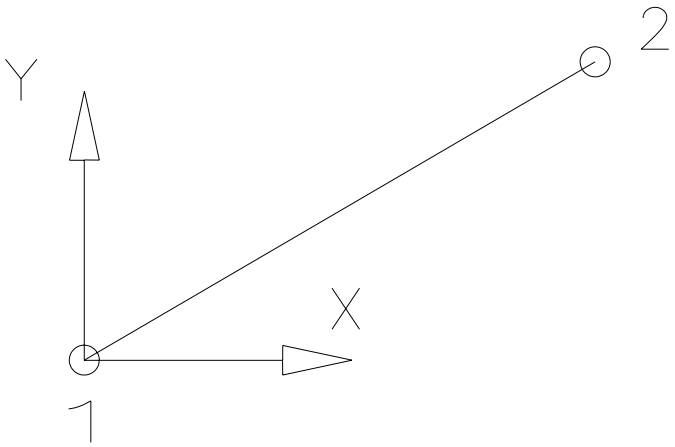
Displacements in R and Z (= X and Y).

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), SIGTE = stress in peripherical direction = tangential stress. Optional von Mises stresses.

Nodal forces in R (= X) and Z (= Y) for each element and each node.

4.9 TRUSS NO.9 IN PLANE

The truss element No.9 can take any location in the X-Y plane. It is the simplest element in Z88 and is calculated extremely fast. The truss elements matches Hooke's law exactly.



Input:

CAD (see chapter 2.7.2): *Line from node 1 to node 2*

Z88I1.TXT

> KFLAG for cartesian (0) or polar coordinates (1)
> 2 degrees of freedom for each node
> Element type is 9
> 2 nodes per element
> Cross-section parameter QPARA is the cross-sectional area of the truss

Z88I3.TXT

Trusses No.9 have no influence. However, Z88I3.TXT must exist (with any content).

Results:

Displacements in X andY

Stresses: Normal stresses

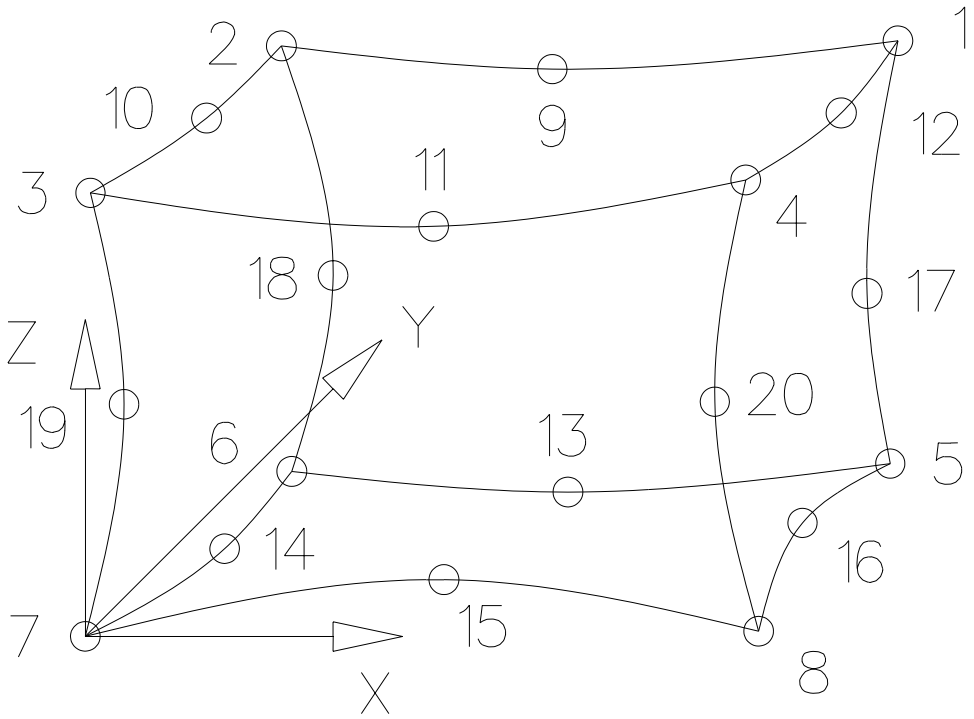
Nodal forces in X and Y for each element and each node.

4.10 HEXAHEDRON NO.10 WITH 20 NODES

This is a curvilinear Serendipity volume element with square shape functions. The transformation is isoparametric. The integration is carried out numerically in all axises according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 is good. This element calculates both displacements and stresses very exactly. The quality of the displacement and stress calculations are far better than the results of the hexahedron element No.1.

Hexahedron No.1 also applies well for thick plate elements, if the plate's thickness is not too small compared to the other dimensions.

The element causes an enormous computing load and needs an extreme amount of memory because the element stiffness matrix has the order 60*60.



The nodal numbering of the element No.10 must be done carefully and must exactly match the sketch below. Pay attention to the location of the axis system ! The possible error message " Jacobi determinant zero or negative " is a hint for incorrect node numbering.

Hexahedron No.10 can be generated by the net generator Z88N from super elements Hexahedron No.10. Thus, the Hexahedron No.10 is well suited as super element. Hexahedron No.10 can also generate 8-node Hexahedrons No.1, see chapter 4.1.

Hexahedron No.10 is recommended for all sort of deflection and stress computation in space. Though its need for memory and computing power is enormous, this element gives precise results for displacements and stresses. Or use it as superelements for meshing Hexahedrons No.1 with 8 nodes.

Input:

CAD (see chapter 2.7.2):
Upper plane: 1 - 9 - 2 - 10 - 3 - 11 - 4 -12 - 1, quit LINE function
Lower plane: 5 - 13 - 6 - 14 - 7 - 15 - 8 - 16 - 5, quit LINE function
1 - 17 - 5, quit LINE function
2 - 18 - 6, quit LINE function
3 - 19 - 7, quit LINE function
4 - 20 - 8, quit LINE function

Z88I1.TXT

> *KFLAG for cartesian (0) or cylindrical coordinates (1)*
> *3 degrees of freedom for each node*
> *Element type is 10*
> *20 nodes per element*
> *Cross-section parameter QPARA is 0 or any value, has no influence*
> *Integration order INTORD for each mat info line. 3 is usually good.*

Z88I3.TXT

> *Integration order INTORD* for stress calculation:
Can be different from INTORD in Z88I1.TXT.
0 = Calculation of stresses in the corner nodes
1,2,3,4 = Calculation of stresses in the Gauss points

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses in the Gauss points (INTORD not 0!)

Results:

Displacements in X, Y and Z

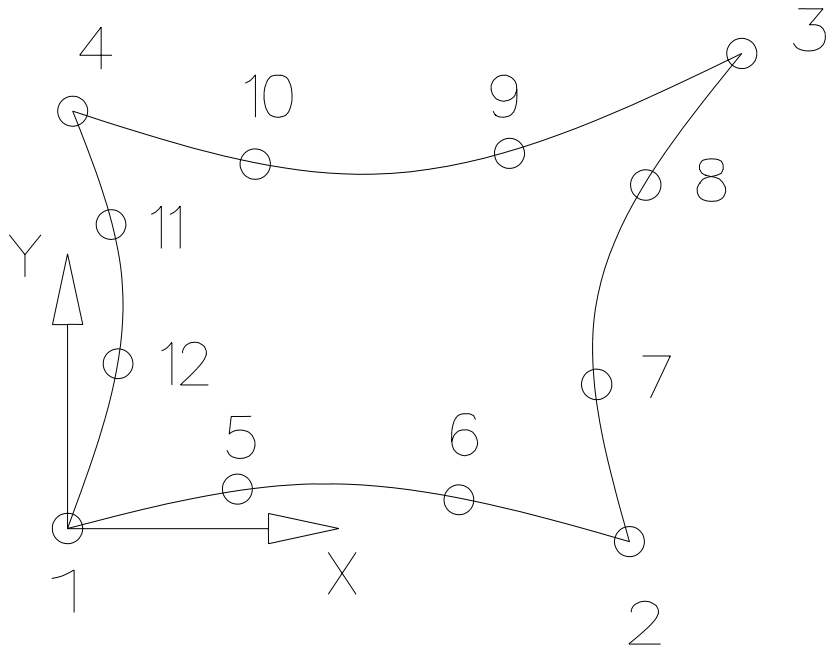
Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises stresses.

Nodal forces in X, Y and Z for each element and each node.

4.11 PLANE STRESS ELEMENT NO.11 WITH 12 NODES

This is a curvilinear Serendipity plane stress element with cubic shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 is mostly the best choice. This element calculates both displacements and stresses with outstanding precision. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Because of its 24*24 element stiffness matrices the element No.11 needs a lot of memory and computing power. Pay attention to edge loads, cf. chapter 3.4.

Plane Stress Elements No.7 can be generated by the net generator Z88N from super elements Plane Stress Elements No.11. Thus, the Plane Stress Element No.11 is well suited as super element. But Plane Stress Elements No.11 cannot be generated by the net generator Z88N from super elements Plane Stress Elements No.11.



Input:

CAD (see chapter 2.7.2): *1-5-6-2-7-8-3-9-10-4-11-12-1*

Z88I1.TXT

> KFLAG for cartesian (0) or polar coordinates (1)
> 2 degrees of freedom for each node
> Element type is 11
> 12 nodes per element
> Cross-section parameter QPARA is the element thickness
> Integration order INTORD per each mat info line. 3 is usually good.

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted
0 = Calculation of the stresses in the corner nodes
1,2,3,4 = Calculation of the stresses in the Gauss points

> *KFLAG* = 0: Calculation of SIGXX, SIGYY and TAUXY
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements ino X and Y.

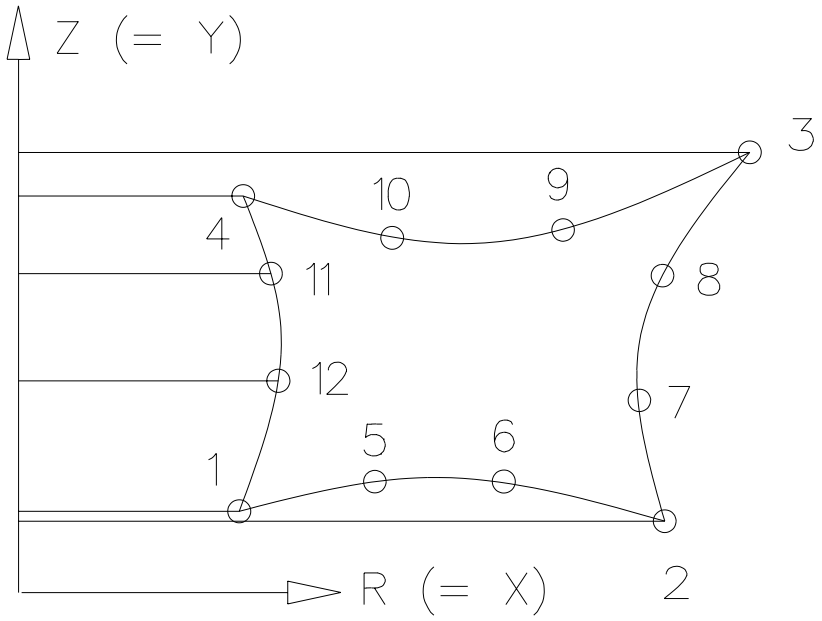
Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. For KFLAG = 1 the radial stresses SIGRR, the tangential stresses SIGTT and the accompanying shear stresses SIGRT are computed additionally (makes only sense if a rotational-symmetric structure is available). For easier orientation the respective radiuses and angles of the nodes/points are printed. Optional von Mises stresses

Nodal forces in X and Y for each element and each node.

4.12 TORUS NO.12 WITH 12 NODES

This is a curvilinear Serendipity torus element with cubic shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss-Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 is mostly sufficient. This element calculates both displacements and stresses with outstanding precision. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Because of its 24*24 element stiffness matrices the element No.11 needs a lot of memory and computing power. Pay attention to edge loads, cf. chapter 3.4.

Torus elements No.8 can be generated by the net generator Z88N from super elements torus elements No.12. Thus, the torus element No.12 is well suited as super element. But torus elements No.12 cannot be generated by the net generator Z88N from super elements torus elements No.12.



Input:

CAD (see chapter 2.7.2): *1-5-6-2-7-8-3-9-10-4-11-12-1*

Z88I1.TXT

> In principle cylindrical coordinates are expected: KFLAG must be 0 !
R coordinate (= X), always positive
Z coordinate (= Y), always positive
> 2 degrees of freedom for each node, DOF R and Z (= X and Y).
> Element type is 12
> 12 nodes per element
> Cross-section parameter QPARA is 0 or any value, no influence
> Integration order per each mat info line. 3 is usually good.

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted
0 = Calculation of the stresses in the corner nodes
1,2,3,4 = Calculation of the stresses in the Gauss points

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

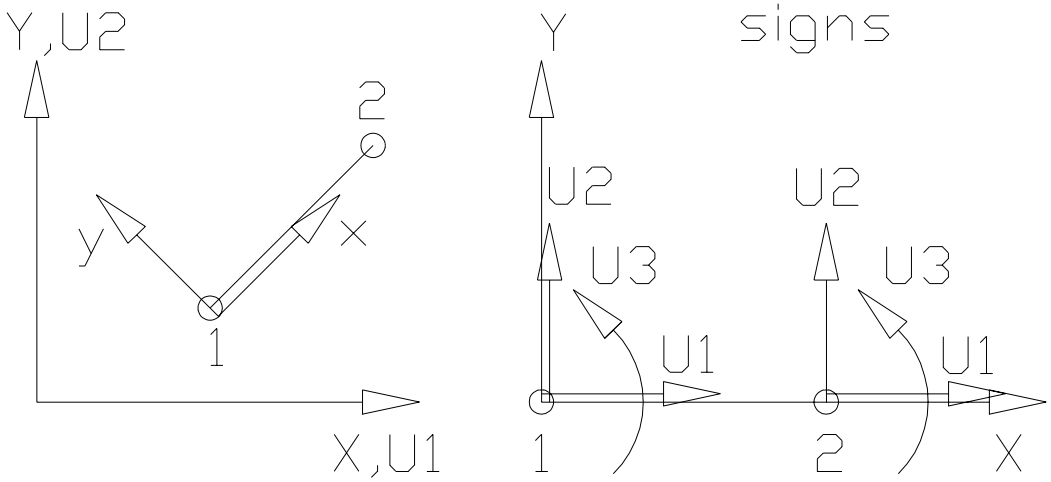
Displacements in R and Z (= X and Y).

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), SIGTE = stress in peripherical direction = tangential stress. Optional von Mises stresses.

Nodal forces in R (= X) and Z (= Y) for each element and each node.

4.13 BEAM NO.13 WITH 2 NODES IN PLANE

Beam element with any symmetric profile. The profile values are provided in Z88I1.TXT. Thus, you can use any symmetric profile in contrast to other FEA programs which sometimes incorporate a variety of different special beam and profile subroutines without matching all symmetric profiles as necessary. The element matches exactly Bernoulli's bend theory and Hooke's law. It uses no approximate solution compared to the continuum elements.



Input:

CAD (see chapter 2.7.2): *Line from node 1 to node 2*

Z88I1.TXT

> KFLAG for cartesian (0) or polar coordinates (1)
> Set beam flag IBFLAG to 1
> 3 degrees of freedom in a node
> Element type is 13
> 2 nodes per element

At the material information lines:
> Any integration order INTORD (1...4), has no influence
> Cross-sectional area QPARA
> Insert 0 for second moment of inertia RIYY (bending around y-y axis)
> Insert 0 for max. distance EYY from neutral axis y-y
> Second moment of inertia RIZZ (bending around z-z axis)
> Max. distance EZZ from neutral axis z-z
> Insert 0 for second moment of area (torsion) RIT
> Insert 0 for second modulus (torsion) WT

Z88I3.TXT

Beams No.13 have no influence. However, Z88I3.TXT must exist (with any content).

Results:

Deflections in X and Y and **rotations** around Z.

Stresses: SIGXX, TAUXX: Direct stress, shear stress, SIGZZ1, SIGZZ2: Bending stress around z-z for node 1 and node 2

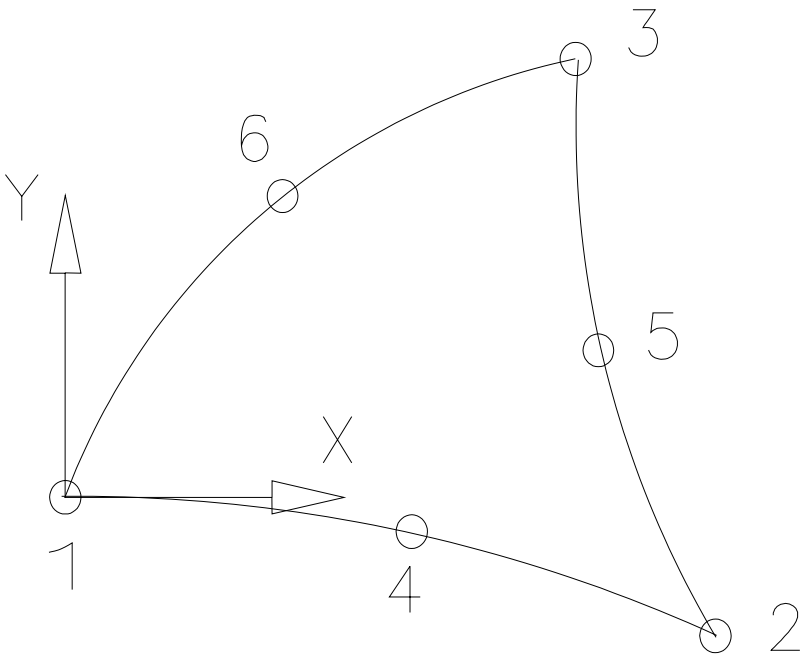
Nodal forces in X and Y and **nodal moments** around Z for each element and each node.

4.14 PLANE STRESS ELEMENT NO.14 WITH 6 NODES

This is a curvilinear Serendipity plane stress element with square shape functions. The transformation is isoparametric. The integration is carried out numerically according to Gauss- Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 7 (=7 Gauss points) is mostly sufficient. This element calculates both displacements and stresses very exactly. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Pay attention to edge loads, cf. chapter 3.4.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a net generation with Z88N is not possible. Use plane stress elements No.7 for Z88N.

Use plane stress element No.7 whenever possible. It is substantially more precise than this isoparametric triangle.



Input:

CAD (see chapter 2.7.2): *1-4-2-5-3-6-1*

Z88I1.TXT

> KFLAG for cartesian (0) or polar coordinates (1)
> 2 degrees of freedom for each node
> Element type is 14
> 6 nodes per element
> Cross-section parameter QPARA is the element thickness

> *Integration order INTORD per each mat info line. 7 is usually good. Possible is: 3 for 3 Gauss points, 7 for 7 Gauss points and 13 for 13 Gausspoints. For easy use with plane stress element No.7 (e.g. with Pro/ENGINEER), function ISOD88 of Z88 uses internally these values:*
integration order 1 or 2 in Z88I1.TXT: 3 Gauss points
integration order 4 in Z88I1.TXT: 7 Gauss points

*Example: Z88I1.TXT uses an entry of 2 for INTORD: Thus, plane stress elements No.7 use 2*2 = 4 Gauss points and plane stress elements No.14 use 3 Gauss points for integration.*

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted

0 = Calculation of the stresses in the corner nodes
1, 7, 13 = Calculation of the stresses in the Gauss points (e.g. 7 Gauss points) See note for Z88I1.TXT.

> *KFLAG* = 0: Calculation of SIGXX, SIGYY and TAUXY
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements ino X and Y.

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. For KFLAG = 1 the radial stresses SIGRR, the tangential stresses SIGTT and the accompanying shear stresses SIGRT are computed additionally (makes only sense if a rotational-symmetric structure is available). For easier orientation the respective radiuses and angles of the nodes/points are printed. Optional von Mises stresses

Nodal forces in X and Y for each element and each node.

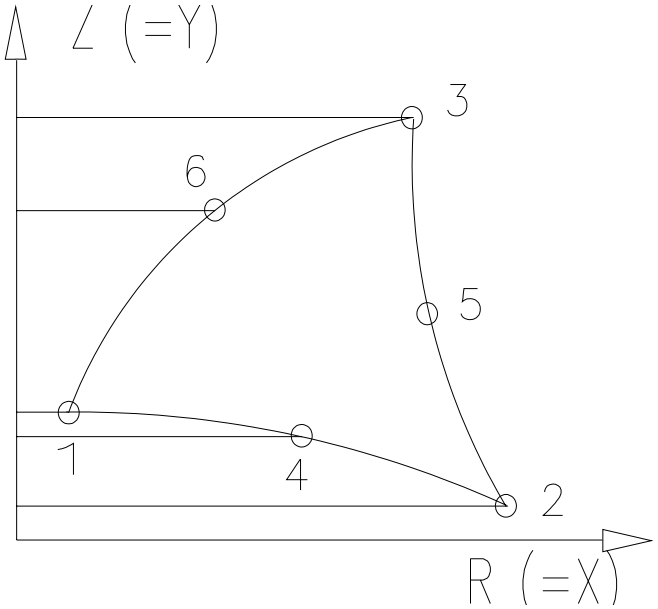
4.15 TORUS NO.15 WITH 6 NODES

This is a curvilinear Serendipity torus element with square shape functions. The transformation is isoparametric. The integration is carried out numerically according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 7 is mostly sufficient. This element calculates both displacements and stresses very exactly. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Pay attention to edge loads, cf. chapter 3.4.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a net generation with Z88N is not possible. Use torus elements No.8 for Z88N.

Use torus element No.8 whenever possible. It is substantially more precise than this isoparametric triangle.

Attention: *This element is not directly integrated into Z88G, because e.g. Pro/MESH for Pro/ENGINEER does not deal at all with these torus elements. But it is easy to overcome this problem: Generate shell in Pro/ENGINEER, lauch Z88G and "find & replace" with an editor the element types No.7 and/or No.14 against element types No.8 and/or No.15. Every better editor has this feature.*



Input:

CAD (see chapter 2.7.2): *1-4-2-5-3-6-1*

Z88I1.TXT

> In principle cylindrical coordinates are expected: KFLAG must be 0 !
R coordinate (= X), always positive
Z coordinate (= Y), always positive
> 2 degrees of freedom for each node, DOF R and Z (= X and Y).

> *Element type is 15*
> *6 nodes per element*
> *Cross-section parameter QPARA is 0 or any value, no influence*
> *Integration order INTORD per each mat info line. 7 is usually good. Possible is: 3 for 3 Gauss points, 7 for 7 Gauss points and 13 for 13 Gausspoints. For easy use with torus element No.8 (e.g. with Pro/ENGINEER), function ISOD88 of Z88 uses internally these values:*
integration order 1 or 2 in Z88I1.TXT: 3 Gauss points
integration order 4 in Z88I1.TXT: 7 Gauss points

*Example: Z88I1.TXT uses an entry of 2 for INTORD: Thus, torus elements No.8 use 2*2 = 4 Gauss points and torus elements No.14 use 3 Gauss points for integration.*

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted

0 = Calculation of the stresses in the corner nodes
1, 7, 13 = Calculation of the stresses in the Gauss points (e.g. 7 Gauss points) See note for Z88I1.TXT.

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements in R and Z (= X and Y).

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), SIGTE = stress in peripherical direction = tangential stress. Optional von Mises stresses.

Nodal forces in R (= X) and Z (= Y) for each element and each node.

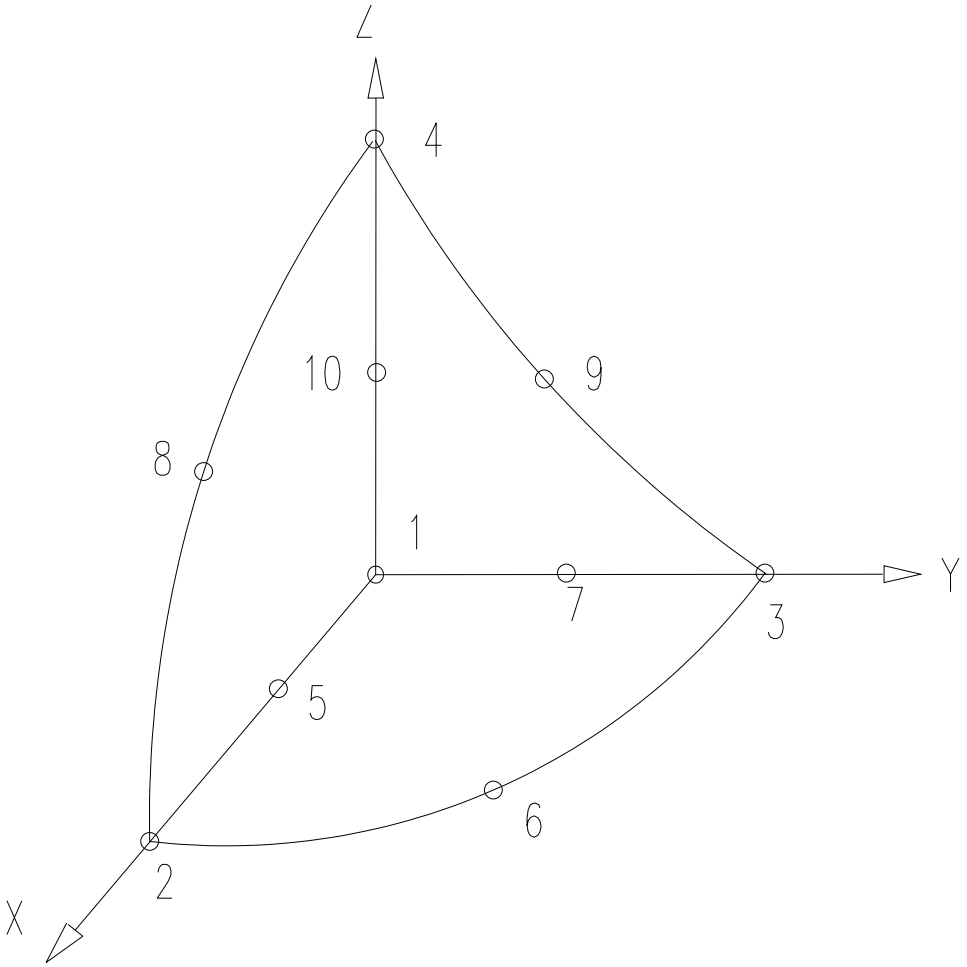
4.16 TETRAHEDRON NO.16 WITH 10 NODES

This is a curvilinear Serendipity volume element with square shape functions. The transformation is isoparametric. The integration is carried out numerically according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 4 is good. The quality of the displacement and stress calculations are far better than the results of the tetrahedron element No.17 but less precise than hexahedron No.10.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesht generation with Z88N and a DXF data exchange with Z88X is not possible, because this will make no sense.

Tetrahedron No.16 also applies well for thick plate elements, if the plate's thickness is not too small compared to the other dimensions.

The element causes a big computing load and needs a large amount of memory because the element stiffness matrix has the order 30*30.



The nodal numbering of the element No.16 must be done carefully and must exactly match the sketch below. Pay attention to the location of the axis system ! The possible error message " Jacobi determinant zero or negative " is a hint for incorrect node numbering.

Tetrahedron No.16 cannot be generated by the net generator Z88N. A DXF data exchange with Z88X is not implemented because tetrahedrons due to their strange geometry are very difficult to arrange in space. This element's main purpose is the use with automeshers from third-party suppliers. **Caution:** Sometimes the automeshers of CAD systems produce very bad element and nodal numbering resulting in an useless large amount of memory needs of Z88F. In this case, renumber especially the nodes.

Input:

Z88I1.TXT

> *KFLAG for cartesian (0) or cylindrical coordinates (1)*
> *3 degrees of freedom for each node*
> *Element type is 16*
> *10 nodes per element*
> *Cross-section parameter QPARA is 0 or any value, has no influence*
> *Integration order INTORD for each mat info line. 4 is usually good. Allowed are 1 for 1 Gauss point, 4 for 4 Gauss points and 5 for 5 Gauss points.*

Z88I3.TXT

> *Integration order INTORD* for stress calculation:
Can be different from INTORD in Z88I1.TXT.
0 = Calculation of stresses in the corner nodes
1, 4, 5 = Calculation of stresses in the Gauss points (e.g. 4 = 4 Gauss points)

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses in the Gauss points (INTORD not 0!)

Results:

Displacements in X, Y and Z

Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises stresses.

Nodal forces in X, Y and Z for each element and each node.

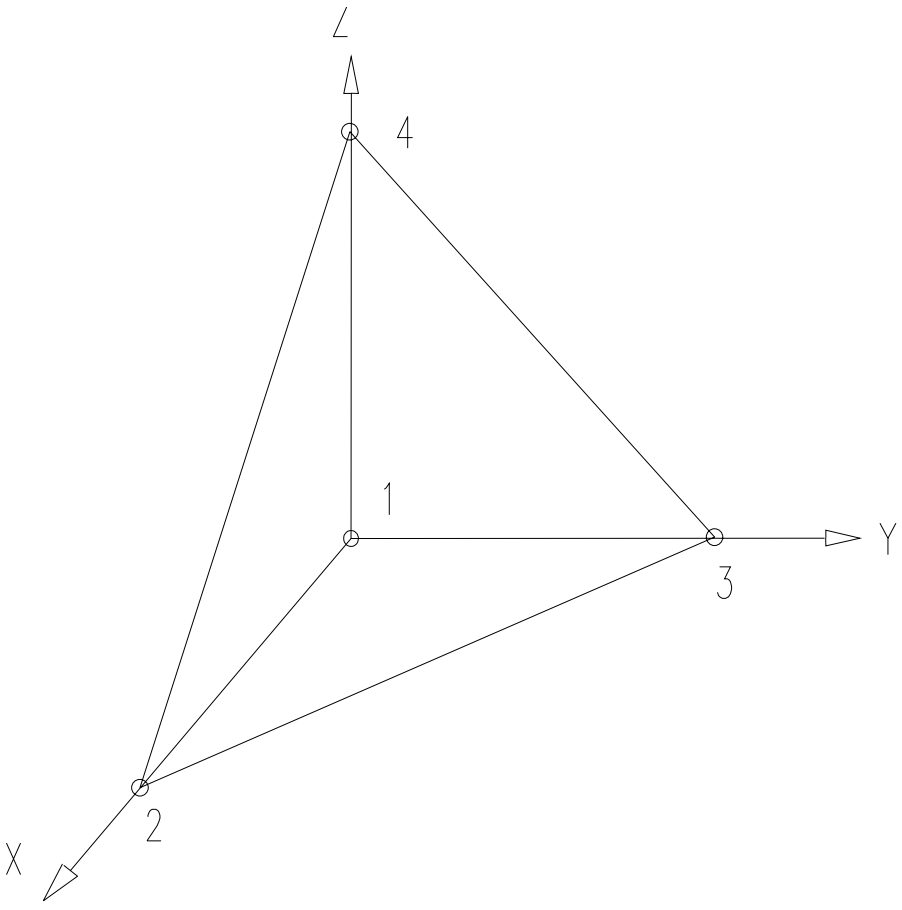
4.17 TETRAHEDRON NO.17 WITH 4 NODES

This is a volume element with linear shape functions. The transformation is isoparametric. The integration is carried out numerically according to Gauss- Legendre. Thus, the integration order can be selected in Z88I1.TXT in the material information lines. The order 1 is good.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a net generation with Z88N and a DXF data exchange with Z88X is not possible, because this will make no sense.

Hexahedron No.1 also applies well for thick plate elements, if the plate's thickness is not too small compared to the other dimensions.

Basically, this element calculates deflections and stresses very bad i.e. inaccurate. One needs very fine meshes to obtain usefull results. Its one and only reason is the data exchange with 3D CAD systems. Use tetrahedrons No.16, hexahedrons No.1 and (best choice) hexahedrons No.10.



Tetrahedron No.17 cannot be generated by the net generator Z88N. A DXF data exchange with Z88X is not implemented because tetrahedrons due to their strange geometry are very difficult to arrange in space. This element's main purpose is the use with automeshers from third-party suppliers. **Caution:** Sometimes the automeshers of CAD systems produce very bad element and nodal numbering resulting in an useless large amount of memory needs of Z88F. In this case, renumber especially the nodes.

Input:

Z88I1.TXT

> *KFLAG for cartesian (0) or cylindrical coordinates (1)*
> *3 degrees of freedom for each node*
> *Element type is 17*
> *4 nodes per element*
> *Cross-section parameter QPARA is 0 or any value, has no influence*
> *Integration order INTORD for each mat info line. 1 is usually good. Allowed are 1 for 1 Gauss point, 4 for 4 Gauss points and 5 for 5 Gauss points.*

Z88I3.TXT

> *Integration order INTORD* for stress calculation:
Can be different from INTORD in Z88I1.TXT.
0 = Calculation of stresses in the corner nodes
1, 4, 5 = Calculation of stresses in the Gauss points (e.g. 4 = 4 Gauss points)

> *KFLAG* , any, has no influence

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = von Mises stresses in the Gauss points (INTORD not 0!)

Results:

Displacements in X, Y and Z

Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises stresses.

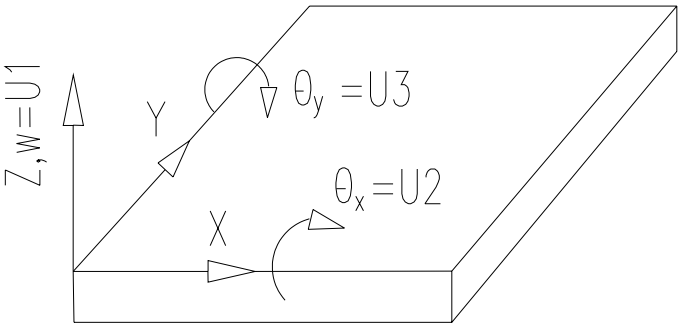
Nodal forces in X, Y and Z for each element and each node.

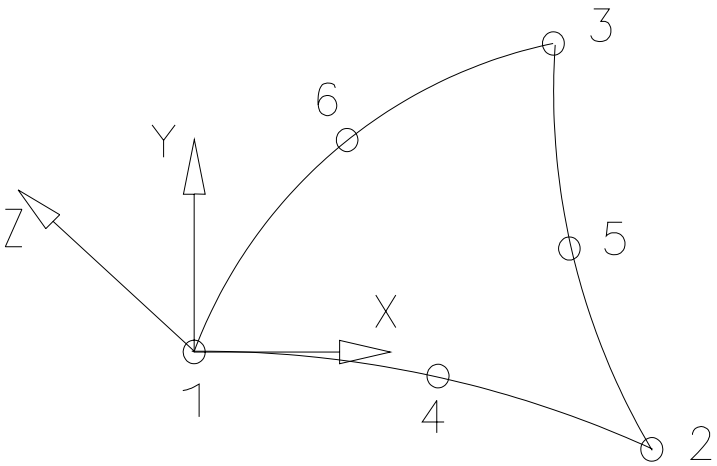
4.18 PLATE NO.18 WITH 6 NODES

This is a curvilinear Serendipity *Reissner- Mindlin* plate element with square shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 3 (= 3 points) is mostly sufficient (reduced integration). This element calculates both displacements and stresses quite good. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Area loads are defined in the appropriate material lines, file Z88I1.TXT, instead of Second moment of inertia RIYY. For this element you need to set the plate flag IPFLAG to 1. Attention: In contrary to the usual rules of the classic mechanics Z88 defines θ_x the rotation around the X- axis and θ_y the rotation around the Y- axis.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesh generation with Z88N is not possible, because this will make no sense. Use plates No.20 for the mesher Z88N.

Because plates No.20 compute both the deflections and the stresses more exactly than the curvilinear triangle plates No.18, you should prefer always plates No.20.





Input:

CAD : *1-4-2-5-3-6-1 ,* ref. chap. 2.7.2

Z88I1.TXT

> KFLAG for cartesian (0) or cylindrical coordinates (1)
> set plate flag IPFLAG to 1 (or 2, if you want to reduce the shear influence)
> 3 degrees of freedom for each node (w, θ_x, θ_y)
> Element type is 18
> 6 nodes per element
> Cross-section parameter QPARA is the element thickness
> "Second moment of inertia RIYY" is the area load
> Integration order INTORD per each mat info line. 3 is usually good. Possible is: 3 for 3 Gauss points, 7 for 7 Gauss points and 13 for 13 Gausspoints. For easy use with plate element No.20 (e.g. with Pro/ENGINEER), function SPLA88 of Z88 uses internally these values:
integration order 1 or 2 in Z88I1.TXT: 3 Gauss points
integration order 4 in Z88I1.TXT: 7 Gauss points

*Example: Z88I1.TXT uses an entry of 2 for INTORD: Thus, plate elements No.20 use 2*2 = 4 Gauss points and plate elements No.18 use 3 Gauss points for integration.*

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted

0 = Calculation of the stresses in the corner nodes
1, 7, 13 = Calculation of the stresses in the Gauss points (e.g. 7 Gauss points) See note for Z88I1.TXT.

> *KFLAG* has no meaning

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = *von Mises* stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements in Z (i.e. w) and rotations θ_x around X- axis and θ_y around the Y- axis.

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. The following results will be presented:

- plate bending moments M_{xx} and M_{yy} (unit: force x lenght / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force x length / length)
- the shear forces Q_{yz} and Q_{zx} (unit: force / length)
- the true stresses resulting from plate bending moments and plate torsion moments

Optional *von Mises* stresses

Nodal forces in X and Y for each element and each node.

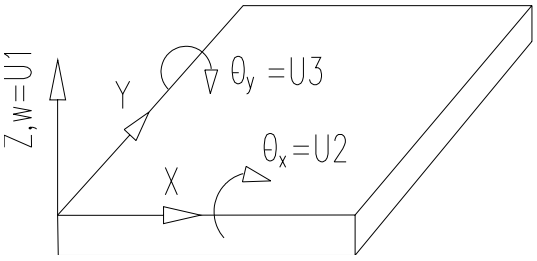
4.19 PLATE NO.19 WITH 16 NODES

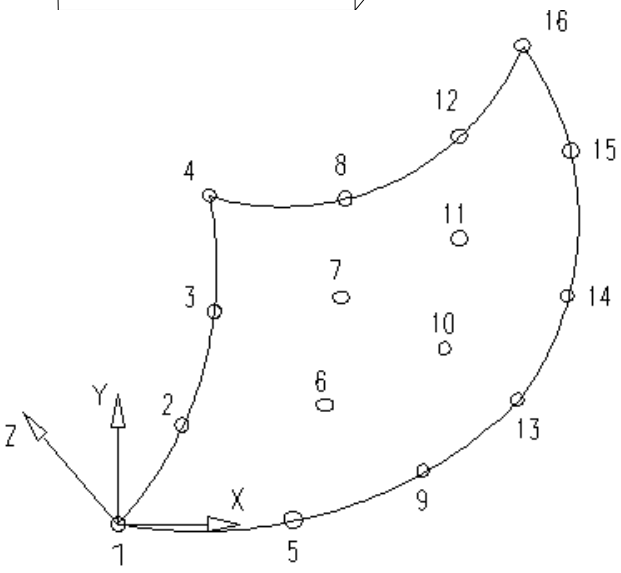
This is a curvilinear Lagrange- *Reissner- Mindlin* plate element with cubic shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 4 (= 4 x 4 points) is very good. This element calculates both displacements and stresses very precisely. The input amount is heavy, you should use the mesher Z88N.

The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Area loads are defined in the appropriate material lines, file Z88I1.TXT, instead of Second moment of inertia RIYY. For this element you need to set the plate flag IPFLAG to 1. Attention: In contrary to the usual rules of the classic mechanics Z88 defines θ_x the rotation around the X- axis and θ_y the rotation around the Y- axis.

Mesh generation with Z88N: Use plates No.20 for super elements, resulting in finite elements of type 19 (plates No.20 may generated by AutoCAD or Pro/ENGINEER, ref. the chapters of Z88X and Z88G). A bit tricky, but works well.
For example, some lines from a mesh generator input file Z88NI.TXT:

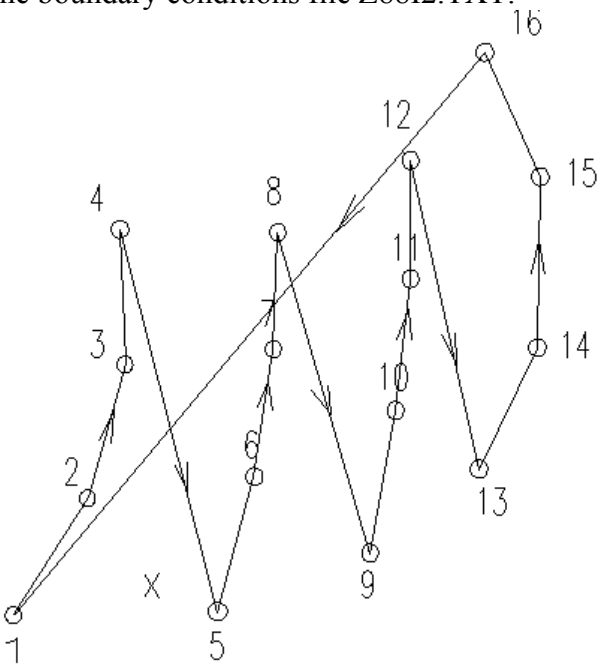
```
.....
5    20          super element 5 of type 20
20   25   27   22   24   26   28   21
.....
5    19          generate from super element 5 (which is of type 20 is, see above) finite elements of type 19
3 E  3 E         .. and subdivide them threetimes equidistant in X-direction and threetimes equidistant in Y- direction
```





Input:

CAD : *1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-1 ,* ref. chap. 2.7.2. Usually, you will not work in this way. It's much more easier to build within a CAD program a super elements mesh with 8- node plates No.20. Export this mesh as a DXF file and use Z88X to produce a mesh generator input file Z88NI.TXT. Run the mesher Z88N and generate a finite elements mesh with plates No.19. Plot this mesh using Z88P, read off the appropriate node numbers and edit the boundary conditions file Z88I2.TXT.



Z88I1.TXT

> *KFLAG for cartesian (0) or cylindrical coordinates (1)*
> *set plate flag IPFLAG to 1 (or 2, if you want to reduce the shear influence)*
> *3 degrees of freedom for each node (w, θ_x, θ_y)*
> *Element type is 19*
> *16 nodes per element*
> *Cross-section parameter QPARA is the element thickness*
> "*Second moment of inertia RIYY" is the area load*
> *Integration order INTORD per each mat info line. 4 is usually good.*

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted

0 = Calculation of the stresses in the corner nodes
1, 2, 3, 4 = Calculation of the stresses in the Gauss points

> *KFLAG* has no meaning

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = *von Mises* stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements in Z (i.e. w) and rotations θ_x around X- axis and θ_y around the Y- axis.

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. The following results will be presented:

- plate bending moments M_{xx} and M_{yy} (unit: force x lenght / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force x length / length)
- the shear forces Q_{yz} and Q_{zx} (unit: force / length)
- the true stresses resulting from plate bending moments and plate torsion moments

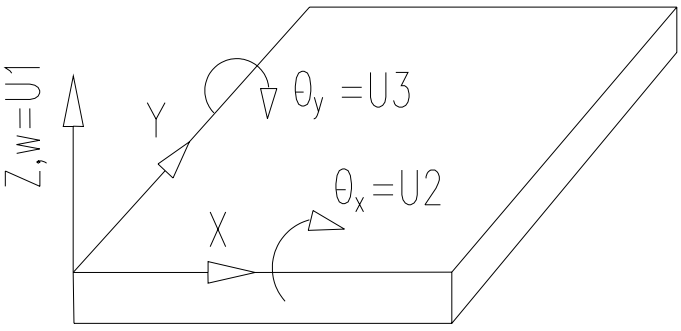
Optional *von Mises* stresses

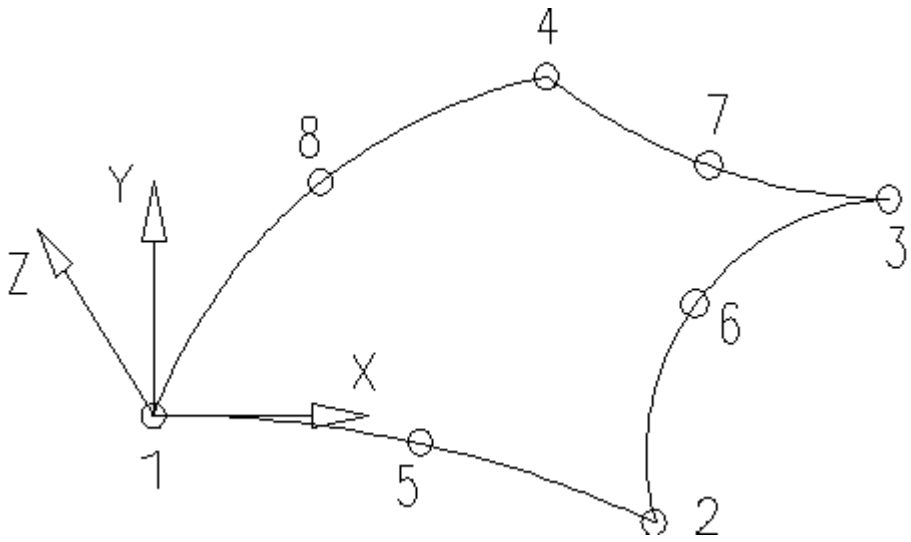
Nodal forces in X and Y for each element and each node.

4.20 PLATE NO.20 WITH 8 NODES

This is a curvilinear Serendipity *Reissner- Mindlin* plate element with square shape functions. The transformation is isoparametric. The integration is carried out numerically in both axises according to Gauss- Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 2 (= 2 x 2 points) is mostly sufficient (reduced integration). This element calculates both displacements and stresses quite good. The integration order can be chosen again for the stress calculation. The stresses are calculated in the corner nodes (good for an overview) or calculated in the Gauss points (substantially more exactly). Area loads are defined in the appropriate material lines, file Z88I1.TXT, instead of Second moment of inertia RIYY. For this element you need to set the plate flag IPFLAG to 1. Attention: In contrary to the usual rules of the classic mechanics Z88 defines θ_x the rotation around the X- axis and θ_y the rotation around the Y- axis.

This element type is implemented for use with automeshers e.g. Pro/MESH for the 3D CAD system Pro/ENGINEER by Parametric Technology. In addition, a mesh generation with Z88N is possible. Super elements of type 20 cannot only generate finite elements of type 20, but plates of type 19, too.





Input:

CAD : *1-5-2-6-3-7-4-8-1,* ref. chap. 2.7.2

Z88I1.TXT

> *KFLAG for cartesian (0) or cylindrical coordinates (1)*
> *set plate flag IPFLAG to 1 (or 2, if you want to reduce the shear influence)*
> *3 degrees of freedom for each node (w, θ_x, θ_y)*
> *Element type is 20*
> *8 nodes per element*
> *Cross-section parameter QPARA is the element thickness*
> "*Second moment of inertia RIYY" is the area load*
> *Integration order INTORD per each mat info line. 2 is usually good.*

Z88I3.TXT

> *Integration order INTORD*: Basically, it is a good idea to use the same value as chosen in Z88I1.TXT , but different values are permitted

0 = Calculation of the stresses in the corner nodes
1, 2, 3, 4 = Calculation of the stresses in the Gauss points

> *KFLAG* has no meaning

> *Reduced stress flag ISFLAG:*
0 = no calculation of reduced stresses
1 = *von Mises* stresses computed for the Gauss points (INTORD not 0 !)

Results:

Displacements in Z (i.e. w) and rotations θ_x around X- axis and θ_y around the Y- axis.

Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. The following results will be presented:

- plate bending moments M_{xx} and M_{yy} (unit: force x lenght / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force x length / length)
- the shear forces Q_{yz} and Q_{zx} (unit: force / length)
- the true stresses resulting from plate bending moments and plate torsion moments

Optional *von Mises* stresses

Nodal forces in X and Y for each element and each node.

5 EXAMPLES

5.0 OVERVIEW

You will find several examples in this chapter along with their respective input files B*.* on the Z88 diskettes. The examples 4, 6 and 7 can be calculated analytically by hand.

Work with the examples which resemble your own applications. Also look at the protocol files *.LOG produced by the Z88 modules. Plot the various examples and results on your plotter or your laser printer capable of HP-GL. Vary the input files, especially the net generator-input files for the examples 1, 5 and 7. Doing so gives you a smart feeling for the howtos of Z88 very quick.

If examples do not run, first search for memory problems. Are there any other programs in the computer's memory, especially those fat and greedy memory eaters like office packages? All examples were tested on various computer equipment and operating systems, and all examples do run even on oldfashioned 386s with 8 MB of RAM along with Windows 95 or the powerful LINUX. Additionally, Z88 is running very large structures on a Silicon Graphics UNIX machine with 6 GB of RAM without any problems. If necessary, adjust Z88.DYN. Investigate the *.LOG files: It is shown here if Z88 modules run out of memory. UNIX: Check file and directory permissions.

After you have investigated the ready-to-run examples, try to draw the examples in your CAD program. Export to DXF files and convert them into Z88 files. If the CAD converter does not convert your DXF files properly, then redo the steps 3 and 5 of chapter 2.6.2, pg. 40 & 41. Did you "snap" the points cleanly ? If nothing works try another CAD program.

Example 1: Fork wrench. Plane stress problem with Serendipity Plane Stress No.7 and net generator use. Learning objectives: CAD and net generator use at curvilinear plane structures, displaying stresses in the plot program. This example is fixed on the Z88 diskette ready to run as the first introduction example with Z88X.DXF, Z88I2.TXT and Z88I3.TXT.

Example 2: Crane truss. Modelled with Trusses No.4. Learning objectives: Use of the different views and rotation possibilities in space within the plot program.

Example 3: Transmission cam. Cam with different diameters, forces and moments in different planes with cam elements No.5, statically overdefined. Learning objectives: Use of the cam elements, especially for the boundary conditions at finite elements with 6 degrees of freedom per node, use of the different views in the plot program.

Example 4: Beam in plane, repeatedly statically overdefined. On both sides firmly fixed Beam No.13. Learning objectives: Use of Beams No.13, choice of the boundary conditions and the interpretation of the results.

Example 5: Disk segment in cake form. General spatial problem with Hexahedrons No.10 (20 nodes) as super elements and net generation of Hexahedrons No.1 (8 nodes). Learning objectives: Use of the net generator at curvilinear spatial elements, showing stresses, different views and spatial rotation possibilities in the plot program. After running this example successfully it is a nice idea to make the net generator generating Hexahedrons No.10 instead of Hexahedrons No.1, what is just a breeze. But you must define new nodes for the boundary conditions.

Example 6: Pipe under inner pressure of 1,000 bar. Axially symmetric problem, solved as plane stress problem with Plane Stress Elements No.7. Learning objectives: Clever use of symmetry qualities of a structure and choice of the proper boundary conditions, showing stresses in the plot program.

Example 7: Press fit. Axially symmetric problem with Tori No.8 and use of net generator. Learning objectives: Work with torus elements, use of the net generator with net compression, stress display in the plot program.

Example 8: Crankshaft. Space structure with Tetrahedrons No.16. Learning objectives: Starting with a COSMOS file from Pro/ENGINEER, we will use the COSMOS converter Z88G, the Cuthill- McKee program Z88H and both the solvers, i.e. the direct Cholesky solver Z88F and the iteration solver Z88I1/Z88I2. This is an example for a larger FEA structure imported from a CAD system.

Example 9: Rectangular plate with 16 nodes Lagrange plate elements No.19. Learning objectives: Starting with an AutoCAD drawing for a super structure with plates No.20, we'll export the DXF file to the CAD converter Z88X. Running the mesh generator Z88N will generate a mesh of plates No.19. The system will be solved by the iteration solver.

Notes:

- The input and output files are printed sometimes shortened to avoid useless pages. Only the essential is shown. You can start every example at any time.

- Remember that 0 (zero) never is real zero but is represented as an approximation to the floating point numbers in a computer. Input values entered in Z88I1.TXT as 0 can appear in output files like Z88O0.TXT as very small numbers which is caused by formattings of the operating system's runtime libraries. This is normal. Of course, this is also true for real calculated results, for example displacements in Z88O2.TXT, stresses in Z88O3.TXT and nodal forces in Z88O4.TXT. Such results have always to be seen in relation to other results: Is, for example, in Z88O2.TXT the biggest calculated displacement 0.1 mm, then consider another displacement, let's say 1.234E-006 mm, as defacto zero.

5.1 FORK WRENCH WITH PLANE STRESS ELE. NO.7

Copy the example files B1_* into Z88 entry files Z88* (has been already carried out on the Z88 diskettes for your immediate start):

B1_X.DXF ---> Z88X.DXF input file for CAD converter Z88X
B1_2.TXT ---> Z88I2.TXT boundary conditions for Cholesky solver Z88F
B1_3.TXT ---> Z88I3.TXT header parameters for stress processor Z88D

Simply proceed with the following steps to get familiar with Z88:

CAD:

As for this first example, you should only look at the CAD super structure without producing it. This comes with later examples. Import Z88X.DXF into your CAD program and view it. Usually you would draw or model the super structure in your CAD system. Do not change anything and leave your CAD program without saving, converting etc. If you do not have any suitable CAD system, then drop this step.

Z88:

Z88X, conversion from Z88X.DXF to Z88NI.TXT. **Windows:** *Compute > Z88X > Type Conversion > 6 from Z88X.DXF to Z88NI.TXT > Compute > Go*, **UNIX:** pushbutton *DXF <-> Z88* with radiobutton *DXF -> NI* (Z88-Commander) or *z88x -nifx* (console or X-term).

Z88P, looking at the super structure**.** The error message should not disturb you, because Z88P still has no .STO file and therefore expects Z88I1.TXT to load (which is not yet present). However, you want to load Z88NI.TXT: **Windows**: *Plot > Z88P > File > Structure File > Z88NI.TXT*, **UNIX:** with the Z88-Commander pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p*, enter *z88ni.txt* in textfield *Struc*., press Return. Now try **Z88O**.

Z88N, mesh generator, reads Z88NI.TXT and produces Z88I1.TXT. **Windows:** *Compute > Z88N > Compute > Go.* **UNIX:** pushbutton *Z88N* (Z88-Commander) or *z88n* (console or X-term).

Z88P, look at the finite element structure. Z88P now has a Z88P.STO file from the former Run and expects Z88NI.TXT to load. Now you want to load Z88I1.TXT. **Windows**: *Plot > Z88P > File > Structure File > Z88I1.TXT*, **UNIX:**, with the Z88-Commander pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p*, enter *z88i1.txt* in textfield *Struc*., press Return. This could have been avoided if you would have deleted the file Z88P.STO before start of Z88P, because Z88P.STO stores all parameters of the last Z88P run. If Z88P.STO does not exist, Z88P works with default values, for example Z88I1.TXT as structure file.

Z88F, calculates displacements. Use the Compactmode: **Windows:** *Compute > Z88F > Mode > Compactmode, > Compute > Go*, **UNIX:** pushbutton *Z88F* with radiobutton *Compact M* (Z88-Commander) or *z88f -c* (console or X-term).

Z88D, calculates stresses. **Windows:** *Compute > Z88D > Compute > Go*, **UNIX:** pushbutton *Z88D* (Z88-Commander) or *z88d* (console or X-term).

Z88P, look at the deflected finite element structure. Z88P has a .STO file and thus features

correct parameters. The displacements are multiplied per default by the factor 100, which is a bit too large for this example. **Windows:** *Plot > Z88P > Factors > Deflections >* enter *10* for *FUX* and *FUY* each, *> Structure > Deflected* **UNIX:** Z88-Commander: pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p*, enter *10* into both textfields *FUX* and *FUY*, press either a Return for each textfield or pushbutton *Regen*. Click radiobutton *Deflected*. Because Z88D has been already run, you can display the von Mises stresses right now. Go to *Undeflected* structure. **Windows:** *> Mises Stresses > Show von Mises stresses*. **UNIX:** Togglebutton *Mises* . Furthermore you could produce a plotter file, perhaps undeflected and without von Mises stresses. **Windows:** *> Output > Plotter*. **UNIX:** Pushbutton *Plot*. The resulting plot file Z88O6.TXT contains HP-GL commands.

Z88E, nodal forces calculation. **Windows:** *Compute > Z88E > Compute > Go*, **UNIX:** pushbutton *Z88E* (Z88-Commander) or enter *z88e* from a console or X-term.

Your task:

A fork wrench should be loaded with the screw's tightness torque. A couple of forces are applied in the wrench's mouth according to the torque and the fixed points are assumed to be at the locations where the mechanic's hand grips the wrench. In fact, these clever boundary conditions are doing the same task as (in reality !) the fixed points in the mouth and the forces applied to the grip, but are much easier to handle.

The fork wrench should be modelled by 7 super elements Plane Stress No.7. The mesh generator should produce 66 finite elements from the super elements. The element thickness is 10 mm each. Mesh generation: Local and global axises are not the same direction in this example: Local x direction at super element 1 defines by the local nodes 1 and 2 which correspond to the global nodes 1 and 3. The local y direction of SE 1 is determined by local nodes 1 and 4 which correspond to the global nodes 1 and 7. Further take into account: Super elements which have a joint side must have an absolutely identical subdivision at this side. Thus, SE 1 and SE 2 share the line 3-4-5: The subdivisions in y direction must be exactly the same. Here 3 subdivisions, respectively.

Now calculate this example as indicated above. After that, one can experiment: Subdivide the SE 7 in Z88NI.TXT as a meaningful variation as follows:

```
7  7                ("Super element 7 is of type 7, i.e. Plane Stress Element No.7")
6  L  3  E ( "Subdivide SE 7 into finite elements Plane Stress No.7 and subdivide into x
              direction 6 times geometrically ascending and in y direction 3 times equidistant")
```

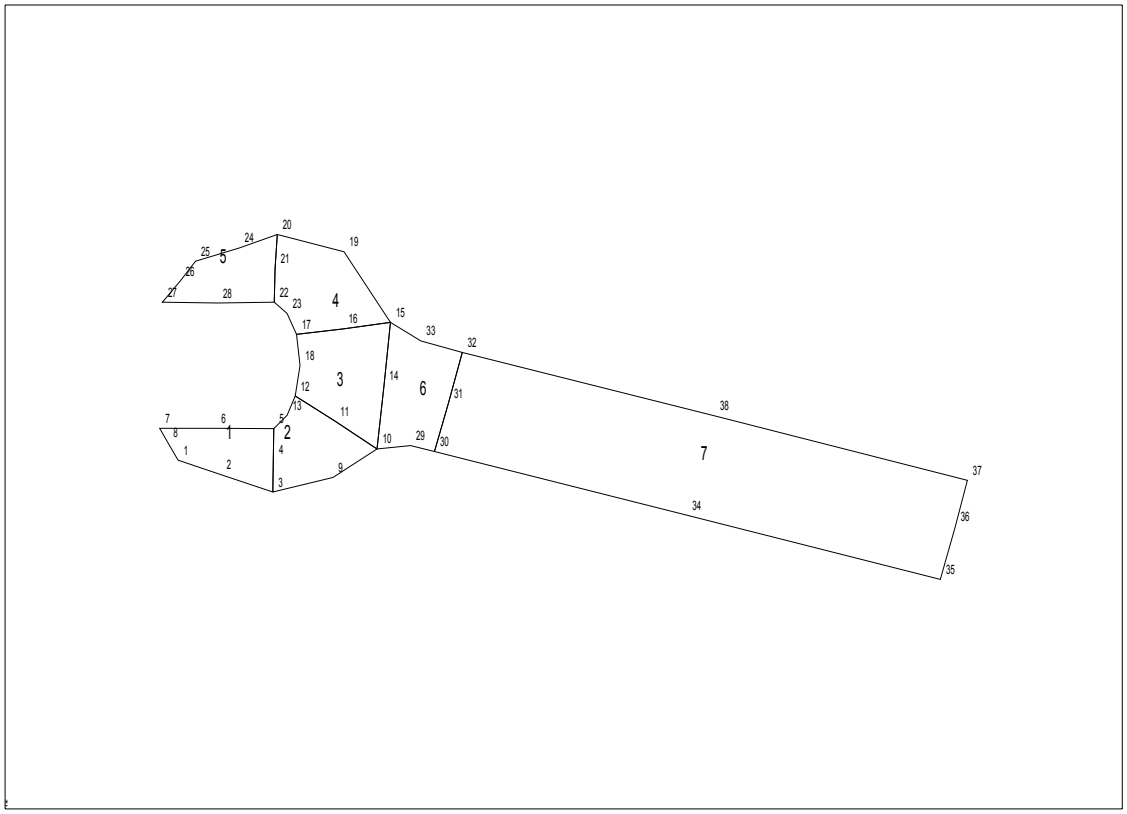
Of course, the SE 1 to SE 5 as well could each be condensed in direction of the screw:

```
1  7
3  L  3  E
2  7
3  L  3  E
.... continue ....
```

Note: As it is obvious for the input files, you can add comments after all required data are entered in every line. Separate the last date from the comment by at least one blank. You can do this just the same in your own files. A maximum of altogether 250 characters per line is permitted.

5.1.1 Input

This example works with a super structure, i.e. a very rough FE mesh. The mesh generator should generate a FE structure from the super structure. Thus, the first task is to design the mesh generator input file Z88NI.TXT. Chapter 2.7 outlines the procedure if working with CAD. If you work without a CAD system, you design the file Z88NI.TXT by editor or word processing program. The super structure shall look as follows:



With CAD program:

Follow the description of chapter 2.7. Do not forget to write the super element information on the layer Z88EIO by TEXT function. Thus
SE 1 7 7 3 E 3 E *(1st SE, SE type7, FE type7, subdiv. x 3 times equid., y 3 times equid.)*
SE 2 7 7 3 E 3 E *(2nd SE, SE type7, FE type7, subdiv. x 3 times equid., y 3 times equid.)*
SE 3 7 7 3 E 3 E
SE 4 7 7 3 E 3 E
SE 5 7 7 3 E 3 E
SE 6 7 7 1 E 3 E
SE 7 7 7 6 E 3 E

...and write the general information and material information onto the layer Z88GEN :
Z88NI.TXT 2 38 7 76 1 0 0 0 0 *(2-DIM,38 nodes,7SE,76 DOF,1 mat info, flags 0)*
MAT 1 1 7 206000 0.3 3 10 *(1.mat info from SE 1 to SE 7: Young's modulus, Poisson's ratio, INTORD, thickness,)*
Export the drawing as DXF file with the name Z88X.DXF and start the CAD converter Z88X with the option "from Z88X.DXF to Z88NI.TXT" (DXF -> NI). Z88X will produce the mesh generator input file Z88NI.TXT. You should have a look at it with Z88P.

With editor:

Write the mesh generator input file Z88NI.TXT (cf. chapter 3.3) with an editor:

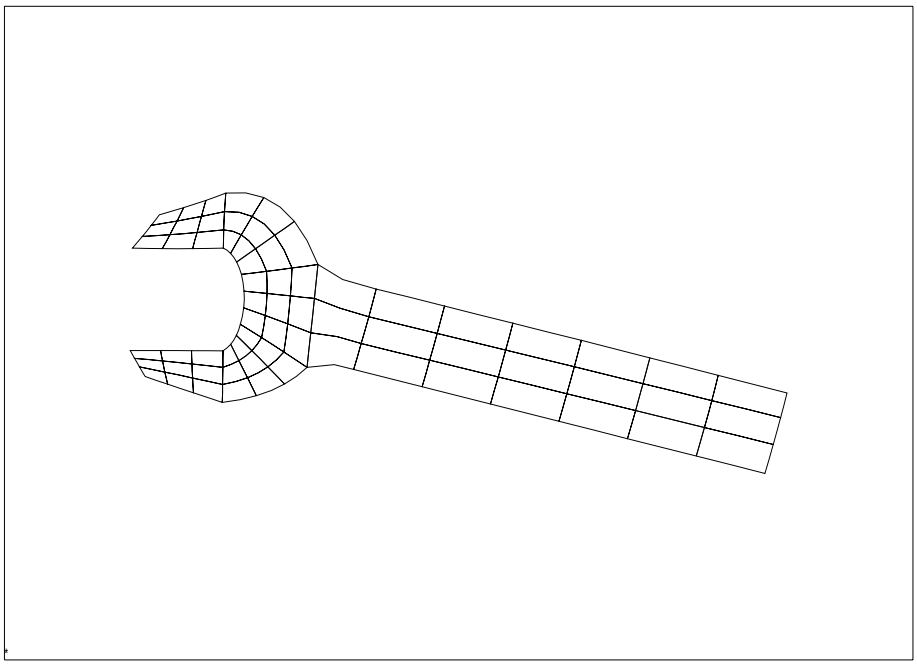
```
2   38   7   76   1   0   0   0   0      (2-DIM,38 nodes,7SE,76 DOF,1 mat info line,flags 0)
1   2   22.040   32.175                  (Node 1, 2 DOF, X and Y coordinates)
2   2   31.913   28.798                  (Node 2, 2 DOF, X and Y coordinates)
3   2   43.781   24.826
4   2   43.880   32.373
5   2   43.980   39.424
 .....                                   (Coordinates for nodes 6... 36 not represented)
37 2   202.847 27.507
38 2   144.905 42.403
1   7                                   (SE 1 of the type Plane Stress No.7)
1   3   5   7   2   4   6   8           (Coincidence for 1st SE)
2   7                                   (SE 2 of the type Plane Stress No. 7)
3 10  12  5   9  11 13  4               (Coincidence for 2nd SE)
.....                                   (Coincidence for elements 3 .. 6 droped here)
7   7
30 35 37 32 34 36 38 31
1  7  206000   0.3   3   10     (mat info from SE 1 to SE 7:Young,Poisson,INTORD,thickness)
1  7                           (Subdivide 1st SE into FE  type 7 and
3  E  3  E                     subdivide into x 3 times equidistant + into y 3 times equidistant)
2  7                           (Subdivide 2nd SE into FE  type 7 and
3  E  3  E                     subdivide into x 3 times equidistant + into y 3 times equidistant)
3  7
3  E  3  E
4  7
3  E  3  E
5  7
3  E  3  E
6  7
1  E  3  E
7  7
6  E  3  E
```

With CAD program and editor:

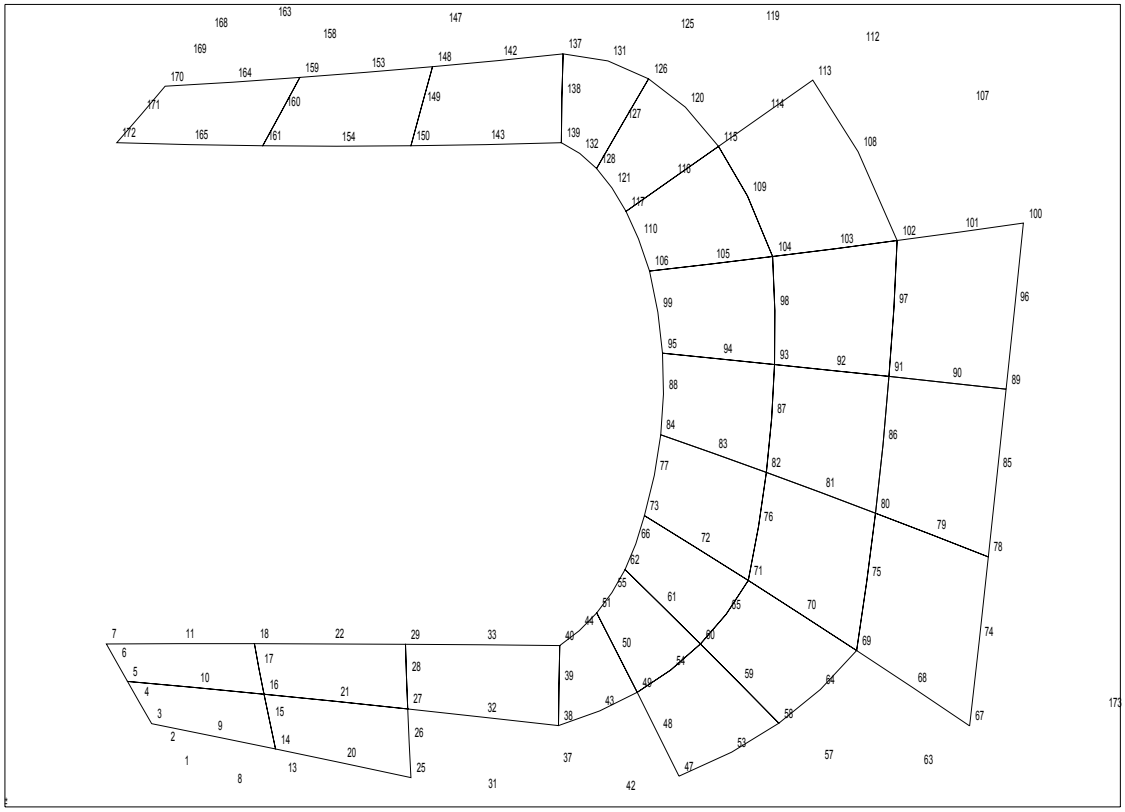
Start the mesh generator Z88N for producing the final Z88 structure file Z88I1.TXT. Look at it either

- in the CAD program (from Z88I1.TXT to Z88X.DXF, I1 -> DXF) after conversion with Z88X or

- with the Z88 plot program Z88P for defining the boundary conditions:



Enlarge the wrench's mouth by zooming for defining the two nodes which will get the load representing the torque (to simplify matters it is assumed, that the screw gets only selectively a couple of forces as torque at the corners and that the screw itself and not the wrench revolves):



We find the nodes 11 and 143. The pictures printed here were produced directly by Z88P.

In the same way both the nodes for fixing the wrench are determined and the boundary conditions are entered in the plot program or CAD system:

In the CAD program:

Switch to the layer Z88RBD and write with the TEXT function into any free place:

```
Z88I2.TXT  16                 (16 Boundary conditions altogether)
RBD   1    11   2   1  -7143  (1st BC: Node 11, DOF 2, Force -7,143 N assumed)
RBD   2   143   2   1   7143  (2nd BC: Node 143, DOF 2, Force 7,143 N assumed)
RBD   3   216   1   2   0     (3rd BC: Node 216, DOF 1, Displacement 0 (= fixed) assumed)
RBD   4   216   2   2   0
RBD   5   220   1   2   0
RBD   6   220   2   2   0
RBD   7   227   1   2   0
RBD   8   227   2   2   0
RBD   9   231   1   2   0
RBD  10   231   2   2   0
RBD  11   238   1   2   0
RBD  12   238   2   2   0
RBD  13   242   1   2   0
RBD  14   242   2   2   0
RBD  15   249   1   2   0
RBD  16   249   2   2   0
```

with an Editor:

Design the boundary condition file Z88I2.TXT by editing:

```
 16                           (16 Boundary conditions altogether)
 11   2   1  -7143            (1st BC: Node 11, DOF 2, Force -7,143 N assumed)
143   2   1   7143            (2nd BC: Node 143, DOF 2, Force 7,143 N assumed)
216   1   2   0               (3rd BC: Node 216, DOF 1, Displacement 0 (= fixed) assumed)
216   2   2   0
220   1   2   0
220   2   2   0
227   1   2   0
227   2   2   0
231   1   2   0
231   2   2   0
238   1   2   0
238   2   2   0
242   1   2   0
242   2   2   0
249   1   2   0
249   2   2   0
```

Input for stress calculation:

In the CAD program:

Switch to the layer Z88GEN and write with the TEXT function into any free place:

Z88I3.TXT 3 0 1 *(3x3 Gauss points for stresses, KFLAG 0, von Mises stresses)*

Export the drawing as DXF file with the name Z88X.DXF, then start the CAD converter Z88X with the option "from Z88X.DXF to Z88I*.TXT" (DXF -> I*). The CAD converter produces the three Z88 input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With an editor:

Enter in the parameter file for the stress processor Z88I3.TXT (cf. Chapter 3.5):

3 0 1 *(3x3 Gauss points for stresses, KFLAG 0, von Mises stresses)*

Now launch the Cholesky solver Z88F and then the stress processor Z88D. You should choose for Z88F the flag *-c* (Compactmode) because only one set of boundary conditions is available cf. section 2.1. Basically, *z88f -c* (Compactmode) is always correct. You will see during the run of Z88F, that 14.848 memory places (8 bytes each) are needed in the total stiffness matrix. NKOI, i.e. memory places in the coincidence vector KOI, is printed as 540 (4 Bytes each). Well, this also matches Z88.DYN. Where does the number 540 come from? 66 finite elements of the type Plane Stress No.7 with 8 nodes each, makes 66*8 = 528. The number 540 results because Z88F always calculates 20 nodes for security reasons for the last finite element. Thus, NKOI becomes here: 65*8 + 20 = 540.

You calculate the nodal forces with Z88E.

5.1.2 Results

The Cholesky solver Z88F provides the following output files:

Z88O0.TXT stores the processed structure data. It is mainly intended for documentation purposes, but also shows if your input file Z88NI.TXT for the mesh generator did what you meant it to do.

Z88O1.TXT stores the processed boundary conditions: For documentation purposes. And: Was your boundary conditions input in Z88I2.TXT correctly interpreted ?

Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

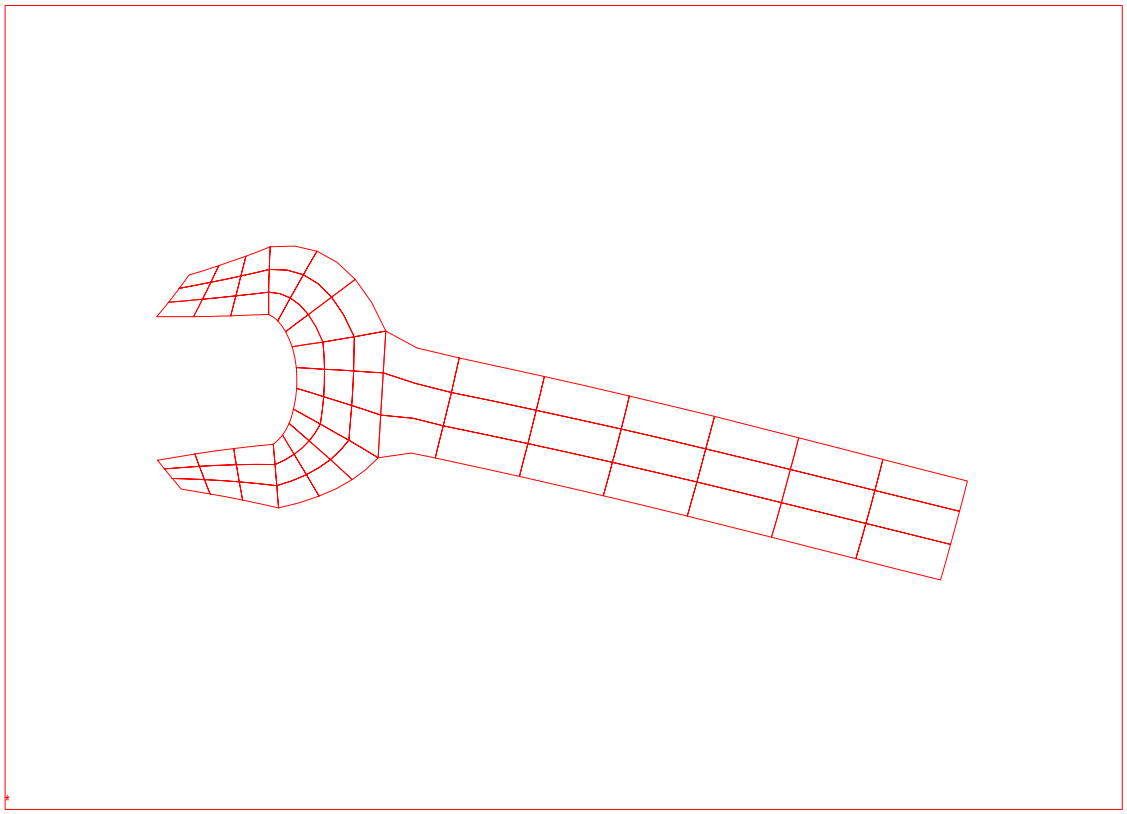
The stress processor **Z88D** uses internally the calculated displacements from Z88F and stores

Z88O3.TXT, the calculated stresses. The results in Z88O3.TXT depend on the header parameters in Z88I3.TXT.

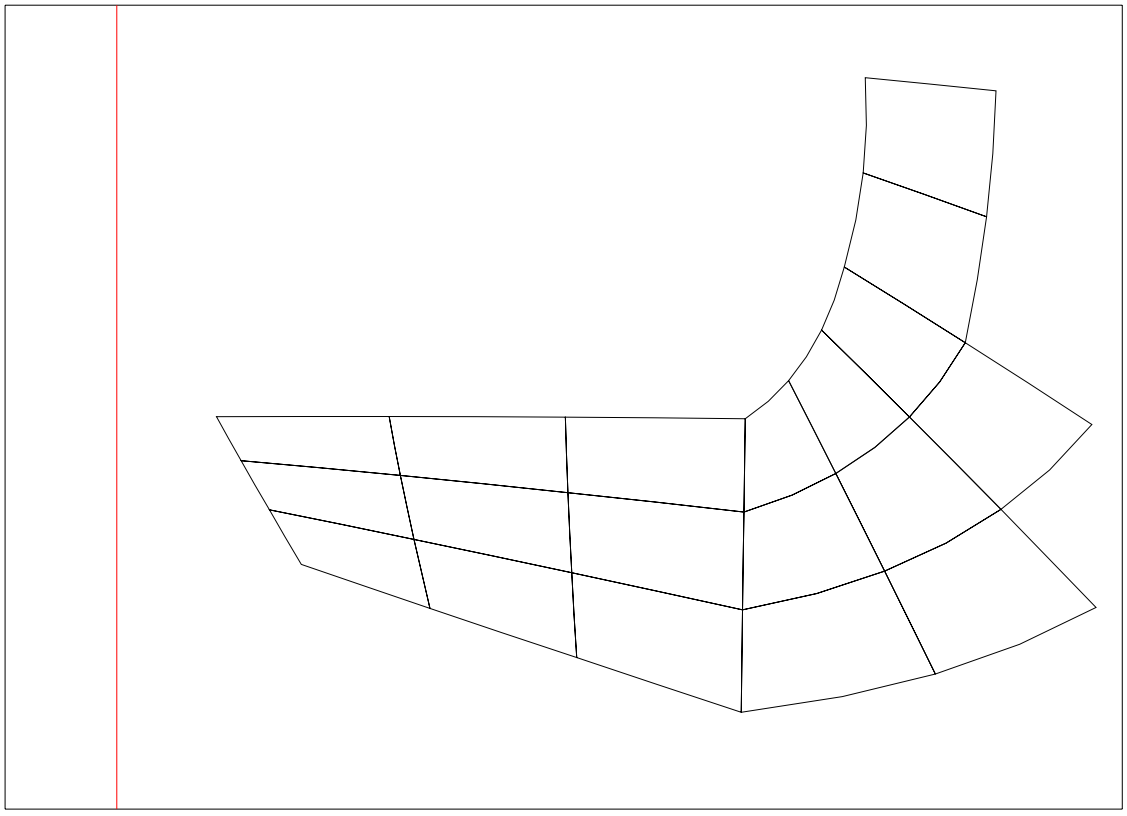
The following picture of the plot program shows the deflected structure for FUX and FUY = 10 each (magnifications of the deflections):

The nodal force processor **Z88E** uses internally the calculated deflections of Z88F and stores

Z88O4.TXT, the computed nodal forces.



Z88P plots the von Mises stresses zoomed around the area of the element 12, which features the biggest von Mises stresses as pointed below. The letter J corresponds to a von Mises stress from 647 to 718 N/mm**2. Now do the same with Z88O but keep in mind that the maximum stresses are somewhat lower, ref. Chp. 2.11.



5.2 CRANE TRUSS WITH TRUSSES NO.4

Copy the example file B2_X.DXF to Z88X.DXF.
B2_X.DXF---> Z88X.DXF input file for CAD converter Z88X

CAD:

In this example you should only look at the CAD FE structure without producing it. This comes with later examples. Import Z88X.DXF into your CAD program and view it. Usually you would draw or model the structure in your CAD system. Do not change anything and leave your CAD program without saving, converting etc. If you do not have any suitable CAD system, then drop this step.

Z88:

Z88X, conversion from Z88X.DXF to Z88I1.TXT, Z88I2.TXT and Z88I3.TXT. **Windows:** *Compute > Z88X > Type Conversion > 5 from Z88X.DXF to Z88I*.TXT* (the default) > *Compute > Go*, **UNIX:** pushbutton *DXF <-> Z88* with radiobutton *DXF -> I** (Z88-Commander) or *z88x -iafx* ("i all from dxf") (console or X-term).

Z88P, look at finite element structure. First delete the file Z88P.STO. Then Z88P loads the structure file Z88I1.TXT per default. **Windows:** *Plot > Z88P*, **UNIX:** with the Z88-Commander pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p.* Now try **Z88O**.

Z88F, calculates deflections. You can use the Compactmode: **Windows:** *Compute > Z88F > Mode > Compactmode, > Compute > Go*, **UNIX:** pushbutton *Z88F* with radiobutton *Compact M* (Z88-Commander) or *z88f -c* (console or X-term).

Z88D, calculates stresses. **Windows:** *Compute > Z88D > Compute > Go*, **UNIX:** pushbutton *Z88D* (Z88-Commander) or *z88d* (console or X-term).

Z88E, nodal forces calculation. **Windows:** *Compute > Z88E > Compute > Go*, **UNIX:** pushbutton *Z88E* (Z88-Commander) or enter *z88e* from a console or X-term.

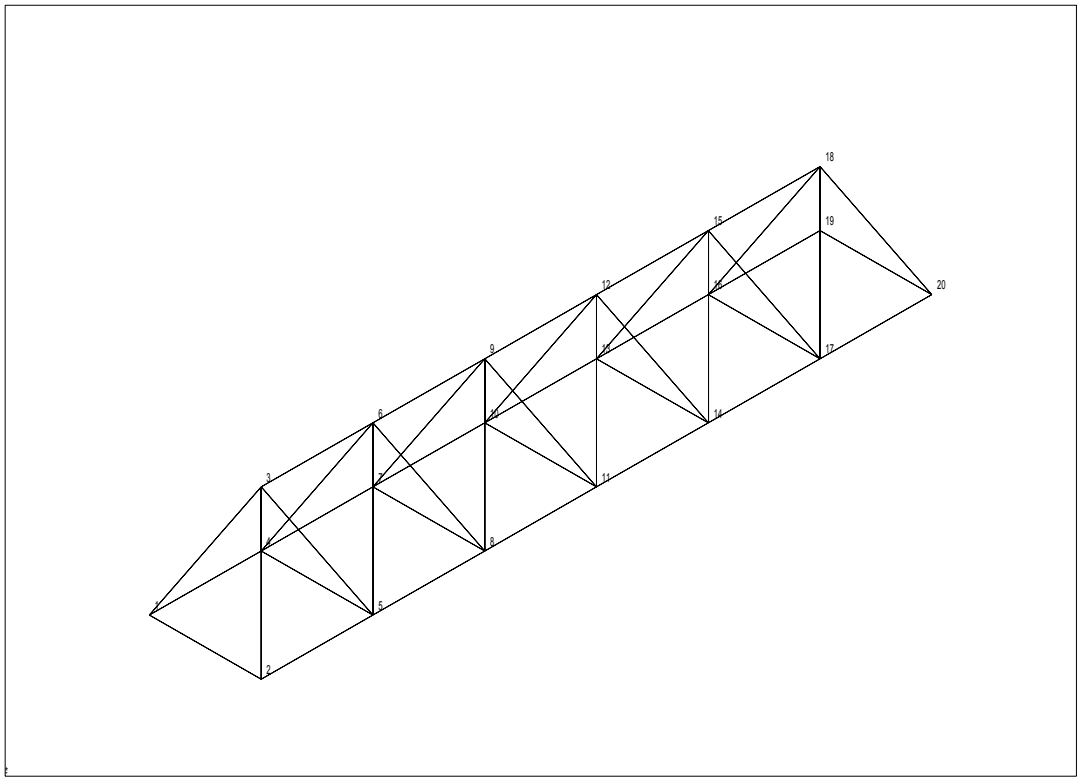
Z88P, look at the deflected finite element structure. The deflections are magnified per default by the factor 100 which is correct for this example. **Windows:** *Plot > Z88P > Structure > Deflected* **UNIX:** Z88-Commander: pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p* click radiobutton *Deflected*. Showing von Mises stresses is not provided for Trusses No.4, because only normal stresses do exist. But with the new Z88O you may plot the tensile stresses as “von Mises stresses”. Have a try!

The example is simple and straight. Experiment with the 3Ds possibilities of the plot programs Z88P and Z88O.

A crane truss consists of 54 trusses, 20 nodes and forms a spatial framework. The nodes 1, 2 and 19, 20 are fixed, the nodes 7 and 8 are loaded per -30,000 N. The total length is 12 m . The inputs in the sample file are in mm but inputs in meters are just as possible if the other entries like Young's modulus and cross-sectional area also refer to meters (or yards or inches). The Young's modulus is 200,000 N/mm*mm, Poisson's ratio 0.3, the cross-sectional area 500 mm**2 each.

This example is taken from the (very good) book SCHWARZ, H.R.: FORTRAN Programme

zur Methode der Finiten Elemente. Teubner Verlag, Stuttgart, Germany 1984.
Take into account: The header file Z88I3.TXT for the stress processor can have any content for Trusses No.4. For mixed frameworks containing hexahedrons and trusses, the entries in Z88I3.TXT do only apply for the hexahedrons.



5.2.1 Input

With CAD program:

Proceed after the description chapter 2.7.2. Do not forget to write on the layer Z88EIO the element descriptions by TEXT function:

FE 1 4 *(1st finite element type 4)*
FE 2 4 *(2nd finite element type 4)*
.....(Information not shown for elements 3 to 53)
FE 54 4 *(54th finite element type 4)*

Write on the layer Z88GEN the general information and material information, like

Z88I1.TXT 3 20 54 60 1 0 0 0 *(3-D,20 nodes,54 ele,60 DOF, 1 mat info, flags 0)*

MAT 1 1 54 200000 0.3 1 500 *(1st mat info from element 1 to element 54,Young's modulus, Poisson's ratio, INTORD (any), QPARA is cross-section area of the trusses)*

Since Trusses No.4 are structure elements (and thus cannot be subdivided like finite elements), the mesh generator cannot be used. You can immediately write the boundary conditions with the TEXT function on the layer Z88RBD: The structure should be fixed to the node 1, 2 and 19, 20. A load of 30,000 N each is applied to the nodes 7 and 8. The load should be applied downward, therefore -30,000 N.

```
Z88I2.TXT   10          (10 boundary conditions altogether)
RBD   1    1   2   2   0   (1st   BC: Node 1, DOF 2, Displacement 0 (=fixed in Y direction)
RBD   2    1   3   2   0   (2nd BC: Node 1, DOF 3, Displacement 0 (=fixed in Z direction)
RBD   3    2   1   2   0   (3rd BC: Node 2, DOF 1, Displacement 0 (=fixed in X direction)
RBD   4    2   3   2   0   (4th BC: Node 2, DOF 3, Displacement 0 (=fixed in Z direction)
RBD   5    7   3   1  -30000   (5th BC: Node 7, DOF 3, load -30,000)
RBD   6    8   3   1  -30000
RBD   7   19   1   2   0
RBD   8   19   3   2   0
RBD   9   20   2   2   0
RBD  10   20   3   2   0
```

... And write on the layer Z88GEN into any free place of your drawing the stress parameters for the stress calculation:

```
Z88I3.TXT   0   0   0   (any stress parameters for Trusses No.4)
```

Export the drawing as DXF file with the name Z88X.DXF and then launch the CAD converter Z88X with the option "from Z88X.DXF to Z88I*.TXT" (DXF -> I*). The CAD converter will produce the input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With an editor:

Enter the structure data into Z88I1.TXT by editor (cf. section 3.2):

```
3  20  54  60  1  0  0  0          (3-dim,20 nodes,54 elements,60 DOF, 1 mat info line,flags 0)
 1   3     0       2000      0     (1st node, 3 DOF, X, Y and Z coordinate)
 2   3     0          0      0     (2nd node, 3 DOF, X, Y und Z coordinate)
 3   3  1000       1000   2000
 4   3  2000       2000      0
 5   3  2000          0      0
.....                             (nodes 6 ..18 dropped here)
19   3  12000      2000      0
20   3  12000         0      0
 1   4                            (1st element, type Truss No.4)
 1   2                            (coincidence 1st element)
 2   4                            (2nd element, type Truss No.4)
 4   5                            (coincidence 2nd element)
 3   4
 7   8
.....                             (elements 4 ..53 dropped here)
54   4
17  19
1  54  200000  0.3  1  500       (mat info from ele 1 to 54,Young's modulus,Poisson's ratio,
                                  INTORD (any), QPARA is cross-section area of the trusses)
```

The structure should be fixed to the node 1, 2 and 19, 20. A load of 30,000 N each is applied to the nodes 7 and 8. The load should be applied downward, therefore -30,000 N. Ref. to 2.4:

```
10                          (10 boundary conditions)
 1   2   2   0              (1st BC: Node 1, DOF 2, Displacement 0 (=fixed in Y direction)
 1   3   2   0              (2nd BC: Node 1, DOF 3, Displacement 0 (=fixed in Z direction)
```

```
 2  1  2  0          (3rd BC: Node 2, DOF 1, Displacement 0 (=fixed in X direction)
 2  3  2  0          (4th BC: Node 2, DOF 3, Displacement 0 (=fixed in Z direction)
 7  3  1  -30000     (5th BC: Node 7, DOF 3, load -30,000)
 8  3  1  -30000
19  1  2  0
19  3  2  0
20  2  2  0
20  3  2  0
```

The parameter file for the stress processor Z88I3.TXT can have any content (cf. sections 3.5 and 4.4), because Gauss points, radial and tangential stresses as well as calculation of the von Mises stresses have no significance for Trusses No.4.

CAD and editor:

Because now the structure data Z88I1.TXT, the boundary conditions Z88I2.TXT and the header file for the stress processor Z88I3.TXT (with any content) do exist, you can launch

>Z88F	Cholesky solver for computing the deflections
>Z88D	stress processor
>Z88E	nodal force processor

5.2.2 Results

The Cholesky solver Z88F provides the following output files:

Z88O0.TXT stores the processed structure data. It is meant for documentation purposes mainly.

Z88O1.TXT stores the processed boundary conditions: For documentation purposes.

Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

The stress processor **Z88D** internally uses the calculated displacements from Z88F and stores

Z88O3.TXT, the calculated stresses. The results in Z88O3.TXT do not depend on the header parameters in Z88I3.TXT for Trusses No.4.

The nodal force processor **Z88E** uses internally the calculated deflections of Z88F and stores

Z88O4.TXT, the computed nodal forces.

The following picture of the plot program shows the deflected structure for FUX, FUY and FUZ = 100 each (magnifications of the deflections):

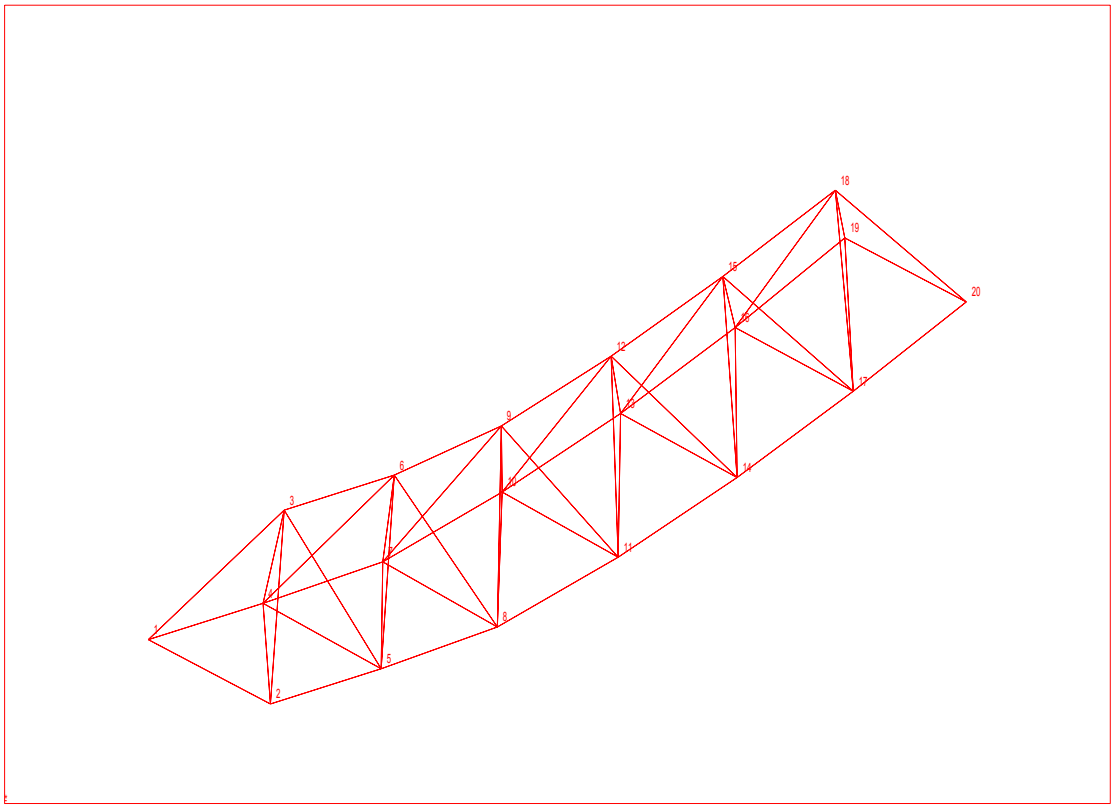
You can rotate the structure of this 3D example under **Windows** with the buttons *F2.. F7* or > *Zoom* > *Rotation X*- etc. or **UNIX** with the Pushbuttons *RX+, RX-,... RZ-* around the three spatial axises in steps of 10 degrees each. *F8* **(Windows)** or *Rot 0* **(UNIX)** resets the rotations to zero.

Do zooming under **Windows** with the keys *PRIOR* or *NEXT* or with > *Zoom* > *More* or > *Zoom* > *Less*. And under **UNIX** with the Pushbuttons *Zoom+* or *Zoom-* .

Do panning under **Windows** with the *cursor keys LEFT, RIGHT, UP, DOWN* and *HOME* or *END*, under **UNIX** with the Pushbuttons *X+, X-, ... Z-*.

In addition, you should try out the different views provided by Z88P: **Windows:** > *View* > *XY, XZ, YZ, 3-Dim* or **UNIX:** Radio box *XY, XZ, YZ, 3D*.

Von Mises stresses are not provided in the plot program for Trusses No.4. But new Z88O does it. Why don't you try it?



5.3 TRANSMISSION CAM WITH CAM ELEMENTS NO.5

Copy the example file B3_X.DXF to Z88X.DXF.
B3_X.DXF---> Z88X.DXF input file for CAD converter Z88X

CAD:

You should only look within this example at the CAD FE structure without producing it. This comes with later examples. Import Z88X.DXF into your CAD program and view it. Usually you would draw or model the structure in your CAD system. Do not change anything and leave your CAD program without saving, converting etc. If you do not have any suitable CAD system, then drop this step.

Z88:

Z88X, conversion from Z88X.DXF to Z88I1.TXT, Z88I2.TXT and Z88I3.TXT. **Windows:** *Compute > Z88X > Type Conversion > 5 from Z88X.DXF to Z88I*.TXT* (the default) > *Compute > Go*, **UNIX:** pushbutton *DXF <-> Z88* with radiobutton *DXF -> I** (Z88-Commander) or *z88x -iafx* ("i all from dxf") (console or X-term).

Z88P, look at finite element structure. First delete the file Z88P.STO. Then Z88P loads the structure file Z88I1.TXT per default. **Windows and UNIX:** You can delete Z88P.STO directly in the Z88 Commander. Then launch the plot program. **Windows:** *Plot > Z88P* **UNIX:** with the Z88-Commander pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p.*

Z88F, calculates deflections. You can use the Compactmode: **Windows:** *Compute > Z88F > Mode > Compactmode, > Compute > Go*, **UNIX:** pushbutton *Z88F* with radiobutton *Compact M* (Z88-Commander) or *z88f -c* (console or X-term).

Z88D, calculates stresses. **Windows:** *Compute > Z88D > Compute > Go*, **UNIX:** pushbutton *Z88D* (Z88-Commander) or *z88d* (console or X-term).

Z88E, nodal forces calculation. **Windows:** *Compute > Z88E > Compute > Go*, **UNIX:** pushbutton *Z88E* (Z88-Commander) or enter *z88e* from a console or X-term.

Z88P, look at the deflected finite element structure. The displacements are multiplied per default by the factor 100, which is a bit too little for this example. **Windows:** *Plot > Z88P > Factors > Deflections >* enter *1000* for *FUX, FUY* and *FUZ* each, *> Structure > Deflected* **UNIX:** with the Z88-Commander pushbutton *Plot feature* and radiobutton *Z88P* or enter from an X-term *z88p.* Enter *1000* into the textfields *FUX, FUY* and *FUZ*, either a Return for each textfield or press pushbutton *Regen*. Click radiobutton *Deflected*.

Basically, the calculation and displaying of von Mises stresses is not provided in Z88 for cams No.5, because newer literal sources state correctly that reduced stresses for cams and other machinery parts under dynamic loads do not only depend on the normal and direct stresses (which are computed by Z88), but also on stress concentration factors (impossible to calculate in Z88 and other FEA systems) and other factors.

Task: A transmission cam is designed as follows:

- Cam section, D= 30 mm, L = 30 mm , fixed bearing at the left end
- Gear wheel 1, reference circle D = 45 mm, L = 20 mm
- Cam section, D = 35 mm, L = 60 mm, moveable bearing in the middle
- Gear wheel 2, reference circle D = 60 mm, L = 15 mm
- Cam section, D = 40 mm, L = 60 mm, moveable bearing at the right end

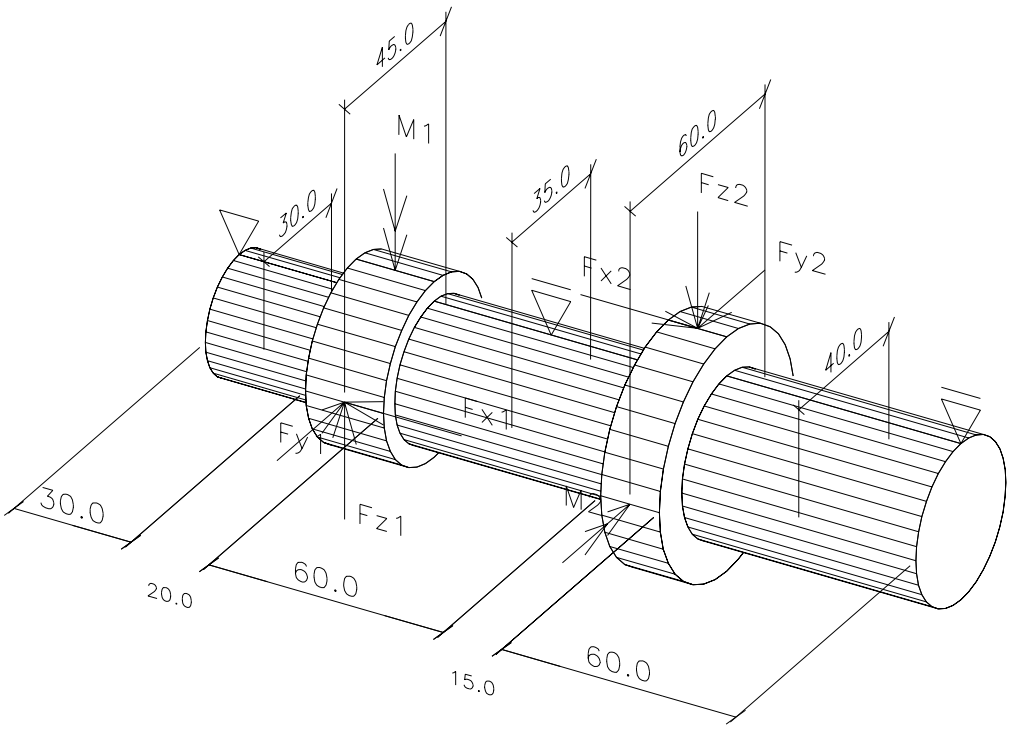
For the loads we picture the cam with the following coordinate system: If we look onto the cam as the main view, then the origin should be at the left end in the middle of the cam. X runs along the cam, Z runs to the upper direction, Y runs in the rear.

Gear wheel 1 gets the following loads in the (physical) point X1 = 40, Y1 = -22.5, Z1 = 0: Fx1 = -10,801 N, Fy1 = 6,809 N, Fz1 = 18,708 N. Fx1 results in a bending moment M1 around the Z axis of -243,023 Nmm.

Gear wheel 2 gets the following loads in the (physical) point X2 = 117.5, Y2 = 0, Z2 = 30: Fx2 = 8,101 N, Fy2 = -14,031 N, Fz2 = -5,107 N. Fx2 results in a bending moment M2 around the Y axis of -243,030 Nmm.

This results in loads in XY and XZ plane. The "physical" points do not exist in the FE calculation, of course, because a cam element is formed analytically only of two points along an axis. The Y and Z coordinates are always 0.

The cam is subdivided into eight cam elements No.5 = 9 nodes. The bearings are assumed in the nodes 1, 5 and 9. Very important: Node 1 is fixed in addition in the degree of freedom 4 (the torsion degree of freedom) in order to compute the torsion angle between the two gears. Otherwise, the structure is statically underdefined !



5.3.1 Input

This example can almost be entered easier by editor into a file than with CAD. The CAD use has real advantages for the examples 1, 2, 5 and 6. Both ways are shown below:

With CAD program:

Proceed according to the description of chapter 2.7. Do not forget to write the element information on the layer Z88EIO by TEXT function:

```
FE  1  5  (1st  finite element type 5)
FE  2  5  (2nd finite element type 5)
FE  3  5  (3rd finite element type 5)
FE  4  5  (4th finite element type 5)
FE  5  5  (5th finite element type 5)
FE  6  5  (6th finite element type 5)
FE  7  5  (7th finite element type 5)
FE  8  5  (8th finite element type 5)
```

Write the general information and material information on the layer Z88GEN :

```
Z88I1.TXT  3  9  8  54  3  0  0  0 (3-Dim, 9 nodes, 8 elements, 54 DOF, 3 mat infos, flags 0 )
MAT 1  1  3 206000 0.3 1 30    (1st  mat info for ele 1 to ele 3, Young's,Poisson's,QPARA)
MAT 2  4  6 206000 0.3 1 35    (2nd mat info for ele 4 to ele 6, Young's,Poisson's,QPARA)
MAT 3  7  7 206000 0.3 1 40    (3rd mat info for ele 7 to ele 7, Young's,Poisson's,QPARA)
                               (INTORD is set here to 1, has no influence)
```

As cam elements No.5 are structure elements (thus not subdividable like finite elements), the mesh generator cannot be used. You can immediately write the boundary conditions with the TEXT function on the layer Z88RBD:

```
Z88I2.TXT  18                (18 Boundary conditions altogether)
RBD  1   1   1   2      0    (1.BC: Node 1, DOF 1 (=X) fixed)
RBD  2   1   2   2      0    (2.BC: Node 1, DOF 2 (=Y) fixed)
RBD  3   1   3   2      0    (3.BC: Node 1, DOF 3 (=Z) fixed)
RBD  4   1   4   2      0    (4.BC: Node 1, DOF 4 (=torsion) fixed)
RBD  5   3   1   1   -10801  (5.BC: Node 3, DOF 1 (=X), load -10,801 N)
RBD  6   3   2   1    +6809  (6.BC: Node 3, DOF 2 (=Y), load 6,809 N)
RBD  7   3   3   1   +18708  (7.BC: Node 3, DOF 3 (=Z), load 18,708 N)
RBD  8   3   4   1   -420930 (8.BC: Node 3, DOF 4 (torsion) -420,930 Nmm)
RBD  9   3   6   1   -243023 (9.BC: Node 3, DOF 6 (bend. moment around Z),-243,023Nmm)
RBD 10   5   2   2      0
RBD 11   5   3   2      0
RBD 12   7   1   1    +8101
RBD 13   7   2   1   -14031
RBD 14   7   3   1    -5107
RBD 15   7   4   1  +420930
RBD 16   7   5   1  -243030
RBD 17   9   2   2      0
RBD 18   9   3   2      0
```

... And write on the layer Z88GEN onto any free place of your drawing the stress parameters

for the stress calculation:

Z88I3.TXT 0 0 0 *(any stress parameters for Trusses No.4)*

Export the drawing as DXF file with the name Z88X.DXF and then launch the CAD converter Z88X with the option "from Z88X.DXF to Z88I*.TXT" (DXF -> I*). The CAD converter will produce the input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With an editor:

Enter the structure data into Z88I1.TXT by editor (cf. section 3.2):

```
3  9  8 54  3   0  0  0      (3D, 9 Node, 8 Ele, 54 DOF, 3 E-Gesetze, Flags 0)
 1  6   0     0  0           (Node 1, 6 DOF, X-, Y- und Z-Koordinate)
 2  6  30     0  0           (Node 2, 6 DOF, X-, Y- und Z-Koordinate)
 3  6  40     0  0
 4  6  50     0  0
 5  6  80     0  0
 6  6 110     0  0
 7  6 117.5   0  0
 8  6 125     0  0
 9  6 185     0  0
1  5                         (Element 1, cam No.5)
1  2                         (Coincidence Ele 1)
2  5                         (Element 2, type 5)
2  3                         (coincidence Ele 2)
..... (Elemente 3 to 7 dropped here)
8  5
8  9
1  3  206000  0.3  1  30     (mat info from Ele 1 to 3,Young's,Poisson's, QPARA= 30)
4  6  206000  0.3  1  35     (mat info from Ele 4 to 6,Young's,Poisson's, QPARA= 35)
7  7  206000  0.3  1  40     (mat info from Ele 7 to 7,Young's,Poisson's, QPARA= 40)
                              INTORD is set here to 1, has no influence.
```

The boundary conditions Z88I2.TXT:

```
18                            (18 Boundary conditions)
 1  1  2       0              (1.BC: Node 1, DOF 1 (=X) fixed)
 1  2  2       0              (2.BC: Node 1, DOF 2 (=Y) fixed)
 1  3  2       0              (3.BC: Node 1, DOF 3 (=Z) fixed)
 1  4  2       0              (4.BC: Node 1, DOF 4 (=torsion) fixed)
 3  1  1   -10801             (5.BC: Node 3, DOF 1 (=X), load -10,801 N)
 3  2  1    +6809             (6.BC: Node 3, DOF 2 (=Y), load 6,809 N)
 3  3  1   +18708             (7.BC: Node 3, DOF 3 (=Z), load 18,708 N)
 3  4  1  -420930             (8.BC: Node 3, DOF 4 (torsion) -420,930 Nmm)
 3  6  1  -243023             (9.BC: Node 3, DOF 6 (bend. moment around Z),-243,023Nmm)
 5  2  2       0
 5  3  2       0
 7  1  1    +8101
 7  2  1   -14031
 7  3  1    -5107
 7  4  1 +420930
 7  5  1  -243030
```

```
 9  2  2      0
 9  3  2      0
```

The parameter file for the stress processor Z88I3.TXT can have any content (cf. sections 3.5 and 4.4), because Gauss points, radial and tangential stresses as well as calculation of the von Mises stresses has no significance for Cam Elements No.5.

CAD and editor:

Because now the structure data Z88I1.TXT, the boundary conditions Z88I2.TXT and the header file for the stress processor Z88I3.TXT (with any content) do exist, you can launch

```
>Z88F       Cholesky solver for computing the deflections
>Z88D       stress processor
>Z88E       nodal force processor
```

5.3.2 Results

The Cholesky solver Z88F provides the following output files:

Z88O0.TXT stores the processed structure data. For documentation purposes.

Z88O1.TXT stores the processed boundary conditions: For documentation purposes.

Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

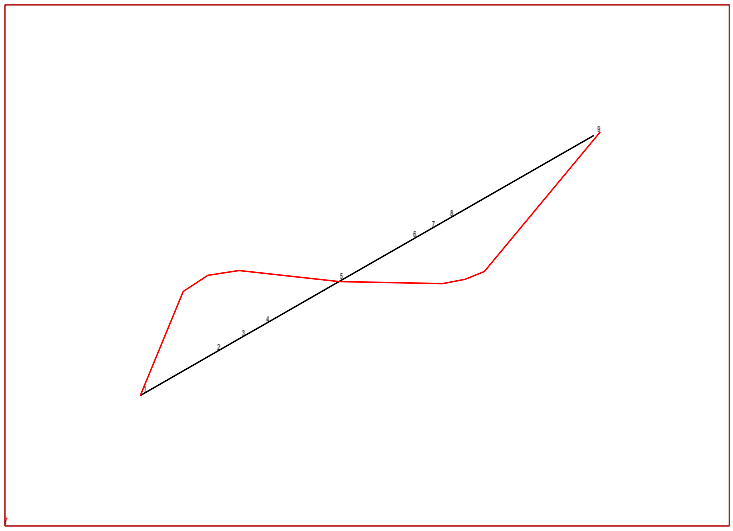
The stress processor **Z88D** internally uses the calculated displacements from Z88F and stores

Z88O3.TXT, the calculated stresses. The results in Z88O3.TXT do not depend on the header parameters in Z88I3.TXT for Cam Elements No.5.

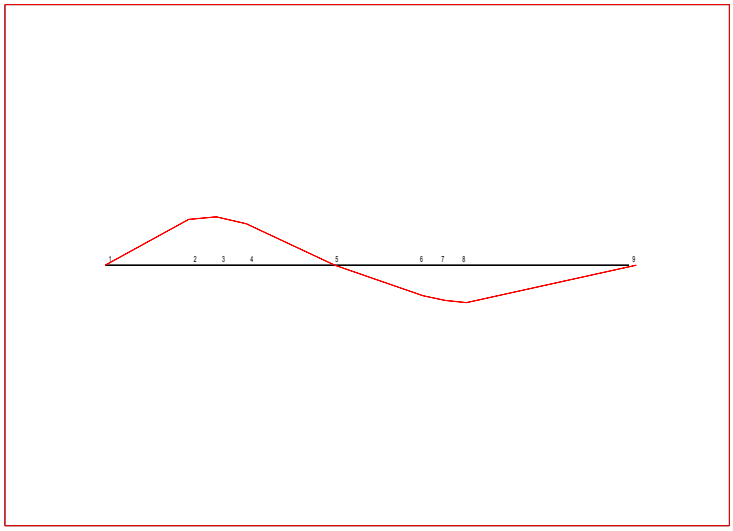
The nodal force processor **Z88E** internally uses the calculated deflections of Z88F and stores

Z88O4.TXT, the computed nodal forces. Keep in mind, that the "forces" of the DOF 4, 5 and 6 are really moments, because the DOF 4, 5 and 6 are rotations.

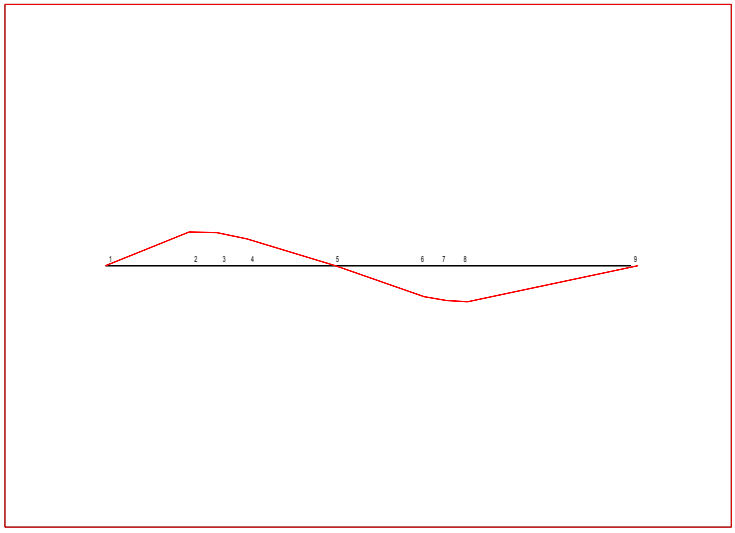
The following pictures of the plot program show the deflected structure for FUX, FUY and FUZ = 1,000 each (magnifications of the deflections):



View of undeflected structure with node labels and deflected structure in space



View of X-Z plane, undeflected and deflected



View of X-Y plane, undeflected and deflected

5.4 BEAM in PLANE WITH BEAMS NO.13

Copy the example file B4_X.DXF to Z88X.DXF.
B4_X.DXF---> Z88X.DXF input file for CAD converter Z88X

CAD:

Import Z88X.DXF into your CAD program and look at it. Usually you would have designed this example in a CAD system (makes not much sense because this example is extremely simple) and then exported as Z88X.DXF.

Z88: (in reduced form, more detailed instructions cf. examples 5.1, 5.2 and 5.3)

Z88X, conversion, "from Z88X.DXF to Z88I*.TXT"
Z88P, looking at structure, structure file Z88I1.TXT
Z88F calculates deflections
Z88D calculates stresses
Z88E calculates nodal forces
Z88P, plot FE structure, now also deflected (FUX, FUY, FUZ per 10.)

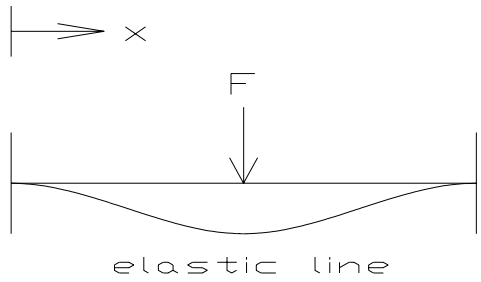
This example deals with a beam, fixed on both sides, and loaded with 1,648 N in the middle in downward direction. This mechanical problem is covered in every mechanical and civil engineering handbook. Geometry: Length 1,000 mm, cross-cut 50 x 10 mm. Thus: A = 500 mm**2, Izz = 4,167 mm**4, ezz = 5 mm.

The deflection curve has inflection points, we therefore take 4 beams No.13. Nodes 1 and 5 will be fixed and node 3 is loaded.

You would calculate analytically:

- *f in the middle: F*L**3/(192*E*I) = 10 mm*
- *f in the inflection points: fw = f/2 = 5 mm*
- *The bending moments on the left, middle, on the right: F*L/8 = 206,000 Nmm*
- *The slope angle in the inflection points: Phi = atan (3*f/L) = 0.029991 rad*

When interpreting the results of Z88O2.TXT (deflections) and Z88O4.TXT (nodal forces and moments) refer to the sign definition of chapter 3.13. Especially Z88O4.TXT, node 3: The force F(2) = force in Y direction is the sum of the forces of elements 2 and 3, due to extrinsic force. The force F(3) = bending moment is not a summary of elements 2 and 3, because it is an intrinsic moment, not an extrinsic load ! Also the signs of the load F(3) at node 1 and F(3) at node 5 are correct, refer to chapter 4.13. Keep in mind that the classical mechanical science sometimes uses different conventions.



5.4.1 Input:

This example shows that a FEA basically needs nodes in all locations where you want to get results. As the beam is fixed left and right, the maximum of displacements appears in the middle for x = L/2, but the bending curve features two inflection points for x = L/4 and x = 3L/4. To calculate results for this locations, the structure must be subdivided with nodes in x = 0, x = L/4, x = L/2 and x = 3L/4.

Only the file input is shown here because CAD use is not worth here.

Z88I1.TXT so becomes:

```
2   5   4  15   1   0   1   0      (2-D, 5 nodes, 4 ele, 5 DOF, 1 mat info, KFLAG 0, IBFLAG 1,
                                    IPFLAG 0)
1   3       0   0                  (1.node, 3 DOF, X and Y coordinate)
2   3     250   0
3   3     500   0
4   3     750   0
5   3    1000   0
1  13                              (1. element, type beam in plane No.13)
1   2                              (coincidence for 1. element)
2  13
2   3
3  13
3   4
4  13
4   5
1   4  206000  0.3  1  500  0  0  4167  5  0  0   (mat info for ele 1 to 4, Young's, Poisson's,
                                                    INTORD (any), QPARA = area, Ixx=0,
                                                    exx=0, Izz, ezz, It=0, Wt=0)
```

The node 1 is fixed in all degrees of freedom at the boundary conditions. It is important to fix especially the DOF 1 = displacement in X direction so that the structure cannot move. Node 5 is fixed in DOF 2 = displacement in Y direction and DOF 3 = rotation around Z axis. You could also fix DOF 1 for node 5, if you wish. But in reality one of the bearings or supports will allow for thermal expansion. This was taken into account in Z88I2.TXT.

Here Z88I2.TXT:

```
6                          (6 Boundary conditions)
 1   1   2     0           (Node 1, DOF 1 gets a displacment of 0 = DOF 1 fixed)
 1   2   2     0           (Node 1, DOF 2 fixed)
 1   3   2     0           (Node 1, DOF 3 fixed (restraining moment)
 3   2   1 -1648           (Node 3, DOF 2 gets load of -1,648 N)
 5   2   2     0
 5   3   2     0
```

The parameter file for the stress processor Z88I3.TXT can have any content (cf. sections 3.5 and 4.13), because Gauss points, radial and tangential stresses as well as calculation of the von Mises stresses has no significance for Beams No.13.

5.4.2 Results

The Cholesky solver Z88F provides the following output files:

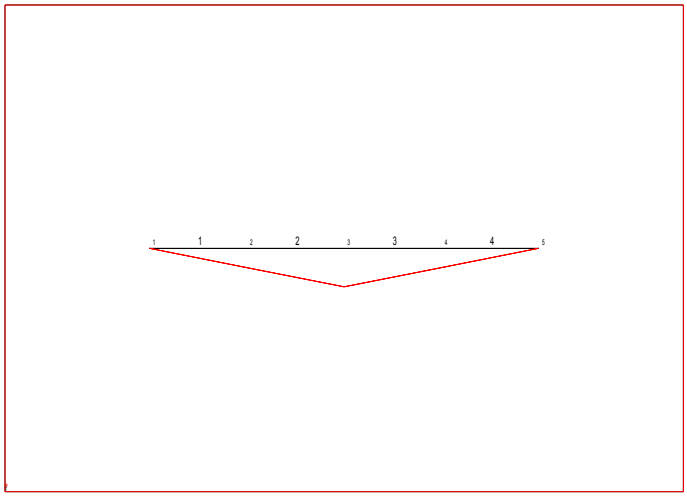
Z88O0.TXT stores the processed structure data. For documentation purposes.
Z88O1.TXT stores the processed boundary conditions: For documentation purposes.
Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

The stress processor **Z88D** internally uses the calculated displacements from Z88F and stores **Z88O3.TXT**, the calculated stresses. The results in Z88O3.TXT do not depend on the header parameters in Z88I3.TXT for Beams No.13.
The nodal force processor **Z88E** internally uses the calculated deflections of Z88F and stores **Z88O4.TXT**, the computed nodal forces.

The following picture of the plot program shows the deflected structure for FUX, FUY and FUZ = 10 each (magnifications of the deflections).

Attention to the results of the nodal force calculation: Node 3: The force F(2) = force in Y direction is the sum of the forces of elements 2 and 3, due to extrinsic force. The force F(3) = bending moment is not a summary of elements 2 and 3, because it is an intrinsic moment, not an extrinsic load ! Also the signs of the load F(3) at node 1 and F(3) at node 5 are correct, refer to chapter 4.13. Keep in mind that the classical mechanical science sometimes uses different conventions.

Additional remark: Such simple examples are well suitable to become aware of the sign omen definitions. Experiment with this example and calculate other bend cases from good handbooks. Frameworks with Beams No.2 are calculated accordingly. However, a real spatial structure then must be available: At least one Z coordinate must not equal 0.



View of undeflected and deflected structure

Take into account: The plot program Z88P connects the nodes with straight lines, although the deflection curve represents a cubic parable in the case of a Beam No.13 or No.2. This means: Z88P shows the deformations correctly for the node, but straight lines are between the nodes. Therefore, no deflection curve is shown. If you want to plot a real nice deflection curve with Z88P, then use basically more nodes, e.g. 15 to 20 nodes for this example (the cubic bending curve is then featured by a couple of straight lines).

5.5 PLATE SEGMENT WITH HEXAHEDRONS NO.1

Copy the example files B5_* into Z88 entry files Z88* :

B5_X.DXF ---> Z88X.DXF input file for CAD converter Z88X
B5_2.TXT ---> Z88I2.TXT boundary conditions for Cholesky solver Z88F
B5_3.TXT ---> Z88I3.TXT header parameters for stress processor Z88D

CAD:

Import Z88X.DXF into your CAD program and look at it. Usually you would have designed this example in a CAD system and then exported it as Z88X.DXF.

Z88: (in reduced form, more detailed instructions cf. examples 5.1, 5.2 and 5.3)

Z88X, conversion, "from Z88X.DXF to Z88NI.TXT"
Z88P, looking at super structure, super structure file Z88NI.TXT
Z88N, computes the finite element mesh
Z88P, looking at finite element structure, structure file Z88I1.TXT, undeflected
Z88X, conversion, "from Z88I*.TXT to Z88X.DXF"

CAD:

Import Z88X.DXF into your CAD program and look at it. Usually you would have now added the boundary conditions and header parameters for Z88I3.TXT and then exported as Z88X.DXF.

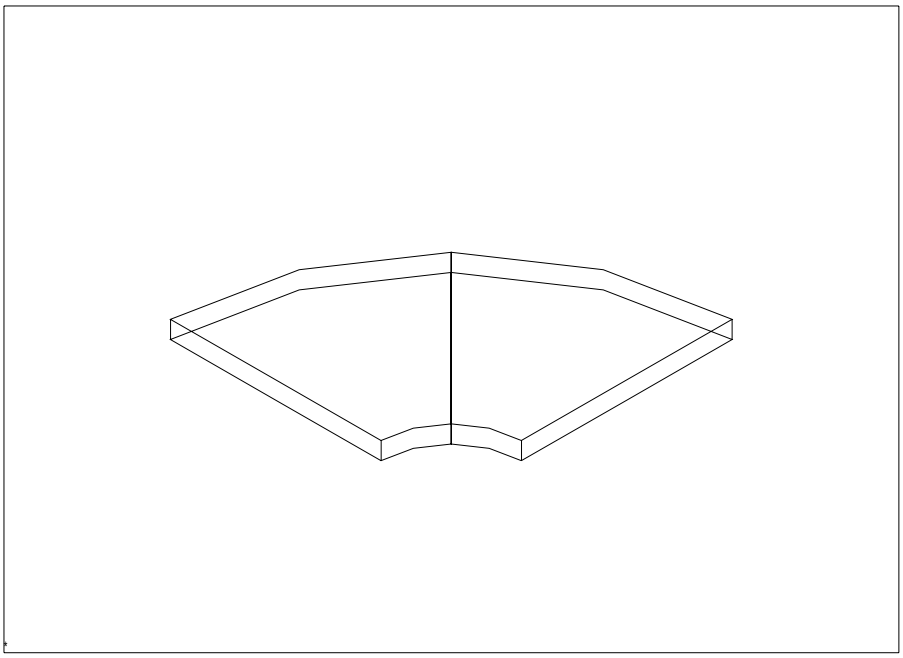
Z88: (in reduced form, more detailed instructions cf. examples 5.1, 5.2 and 5.3)

Z88X, conversion, "from Z88X.DXF to Z88I*.TXT"
Z88F calculates deflections
Z88D calculates stresses
Z88P, plot FE structure, now also deflected (FUX, FUY, FUZ per 10.), show v. Mises stresses
Z88E calculates nodal forces

We deal with a 90 degrees disk segment which looks like a piece of tart. It is fixed at the outer edge and is loaded with 7,000 N at the inner edge. For such structures data entry is best by cylindrical coordinates. To fix the geometry two super elements Hexahedrons No.10 will do fine. These two SE are now to be subdivided into 48 Hexahedrons No.1 for the FE mesh.

This example is very suitable for experiments with the mesh generator . . if you do this, you have to define new boundary conditions, if necessary: With the help of your CAD program or the Z88-plot program.

Concerning the stress indication take into account that the stresses are plotted in the Gauss points. Gauss points lie within of a finite element, never directly on the surface. One gets stresses on the surface by extrapolation, e.g. bending stresses by use of the geometric analogy.



Super structure,consisting of two Hexahedrons No.10 with 20 nodes each

5.5.1 Input

With CAD program:

Use the description in chapter 2.7.2. Do not forget to write the super element information on the layer Z88EIO by TEXT function. Thus

```
SE  1  1  8  L  3  e  1  e (1st super element, finite element type 1, subdivide into x 8 times
                            increasing, into y 3 times equid., no subdivision into z)
SE  2  1  8  L  3  e  1  e (2nd super element, finite element type 1, subdivide into x 8 times
                            increasing, into y 3 times equid., no subdivision into z)
```

Write the general information and material information on the layer Z88GEN:

```
Z88NI.TXT  3   32   2   96   1   1   0   0   0  (3-Dim, 32 nodes, 2 SE, 96 DOF, 1 mat info line,
                                                  KFLAG 1, rest of flags is 0 )
MAT   1   1   2   206000   0.3   2   0          (1st mat info: SE1 to SE2: Young's, Poisson's,
                                                  INTORD for FE, QPARA is 0 )
```

Export the drawing as DXF file with the name Z88X.DXF and start the CAD converter Z88X with the option "from Z88X.DXF to Z88NI.TXT" (DXF -> NI). Z88X will produce the mesh generator input file Z88NI.TXT. (You should have a look at it with Z88P).

With editor:

Write the mesh generator input file Z88NI.TXT (cf. chapter 3.3) with an editor:

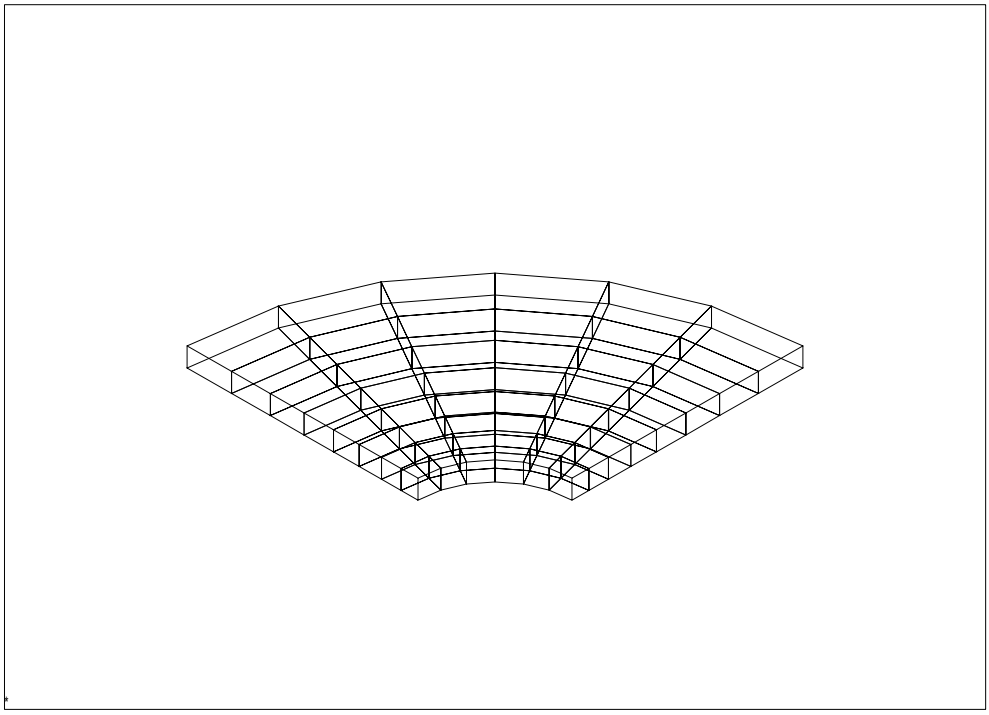
```
 3   32   2   96   1   1   0   0   0 (3-Dim, 32 nodes, 2 SE, 96 DOF, 1 mat info line,
                                       KFLAG 1, rest of flags is 0 )
 1   3   20   0   5                   (1st node, 3 DOF, R-, Phi and Z coordinate)
 2   3   80   0   5                   (2nd node, 3 DOF, R-, Phi and Z coordinate)
 3   3   80  45   5
.....                                  (nodes 4..  30 not represented)
31   3   80  90  2.5
```

```
32  3   20  90  2.5
1   10                                                    (Super ele 1, type Hexah. No.10)
1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20   (coincidence for SE 1)
2   10                                                    (Super ele 2, type Hexah. No.10)
4 3 21 22 8 7 23 24 11 25 26 27 15 28 29 30 20 19 31 32   (coincidence for SE 2)
1   2   206000   0.3   2   0   (SE1 to SE2: Young's, Poisson's, INTORD for FE, QPARA is 0)
1   1                          (Subdivide SE1 into Hexahedrons No.1 and subdivide into
8   L   3   E   1   E          x 8 times increasing, into y 3 times equid., no subdivision into z)
2   1                          (Subdivide SE2 into Hexahedrons No.1 and subdivide into
8   L   3   E   1   E          x 8 times increasing, into y 3 times equid., no subdivision into z)
```

CAD and editor:

Start the mesh generator Z88N to produce the final Z88 structure file Z88I1.TXT. Look at it either

- in the CAD program (from Z88I1.TXT to Z88X.DXF) after conversion with Z88X or
- with the Z88 plot program Z88P for defining the boundary conditions:



View of the FE mesh Z88I1.TXT produced by the mesh generator

Now determine in the plot program or CAD system the nodes which are to be fixed or to be loaded and enter the boundary conditions:

In the CAD program:

Switch to the layer Z88RBD and write with the TEXT function into any free place:

```
Z88I2.TXT   49                  (49 boundary conditions altogether)
RBD   1     1   3   1   -1000   (1st BC: Node 1, DOF 3 (=Z), a load of 1,000 N downward)
RBD   2     3   3   1   -1000
RBD   3     5   3   1   -1000
RBD   4     7   3   1   -1000
RBD   5    65   1   2   0      (5th BC: Node 65, DOF 1 fixed)
```

```
RBD   6   65  2  2  0        (6th BC: Node 65, DOF 2 fixed)
RBD   7   65  3  2  0        (7th BC: Node 65, DOF 3 fixed)
.....(the nodes 66,67,68,69,70,71,72 are fixed in all 3 degrees of freedom, like node 65)
RBD  29  73  3  1  -1000
RBD  30  75  3  1  -1000
RBD  31  77  3  1  -1000
.... (the nodes 121,122,123,124,125 are fixed in all 3 degrees of freedom, like node 126)
RBD 47 126  1  2  0
RBD 48 126  2  2  0
RBD 49 126  3  2  0
```

With editor:

Design the boundary conditions file Z88I2.TXT by editing:

```
 49                  (49 boundary conditions altogether)
  1  3  1  -1000     (1st BC: Node 1, DOF 3 (=Z), a load of 1,000 N downward)
  3  3  1  -1000
  5  3  1  -1000
  7  3  1  -1000
 65  1  2  0         (5th BC: Node 65, DOF 1 fixed)
 65  2  2  0         (6th BC: Node 65, DOF 2 fixed)
 65  3  2  0         (7th BC: Node 65, DOF 3 fixed)
.....(the nodes 66,67,68,69,70,71,72 are fixed in all 3 degrees of freedom, like node 65)
 73  3  1  -1000
 75  3  1  -1000
 77  3  1  -1000
.... (the nodes 121,122,123,124,125 are fixed in all 3 degrees of freedom, like node 126)
126  1  2  0
126  2  2  0
126  3  2  0
```

Input for stress calculation:

With CAD program:

Switch to the layer Z88GEN and write into any free place:

```
Z88I3.TXT   2   0   1    (2x2 Gauss points for stresses, KFLAG 0, von Mises stresses)
```

Export the drawing as DXF file with the name Z88X.DXF, then start the CAD converter Z88X with the option "from Z88X.DXF to Z88I*.TXT" (DXF -> I*). The CAD converter produces the three Z88 input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With editor:

Enter the parameter file for the stress processor Z88I3.TXT (cf. Chapter 3.5):

```
2   0   1                ( 2x2 Gauss points for stresses, KFLAG 0, von Mises stresses)
```

CAD and editor:

Now launch the Cholesky solver Z88F and then the stress processor Z88D. You should choose the Compactmode for Z88F because only one set of boundary conditions is available, cf. section 2.1. Compute nodal forces with Z88E.

5.5.2 Results

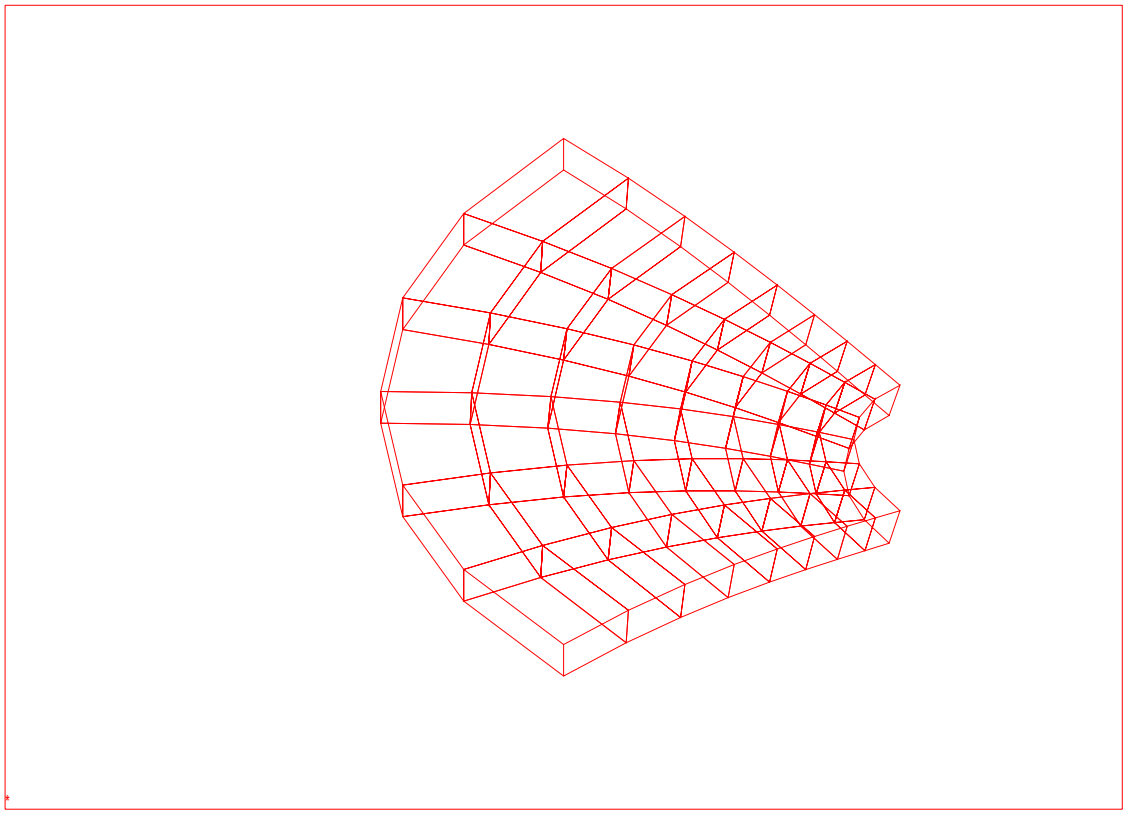
The Cholesky solver Z88F provides the following output files:

Z88O0.TXT stores the processed structure data. For documentation purposes.
Z88O1.TXT stores the processed boundary conditions: For documentation purposes.
Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

The stress processor **Z88D** internally uses the calculated displacements from Z88F and stores **Z88O3.TXT**, the calculated stresses. The results in Z88O3.TXT depend on the header parameters in Z88I3.TXT.

The nodal force processor **Z88E** internally uses the calculated deflections of Z88F and stores **Z88O4.TXT**, the computed nodal forces.

The following picture of the plot program shows the deflected structure for FUX, FUY and FUZ = 10 each (magnifications of the deflections):



Hint: The super structure is very easy to design with e.g. AutoCAD. Draw the edges using arcs. The nodal points can easily be produced by the function > Draw > Point > Divide. **When outlining the elements using the LINE function be sure to position the view in space exactly to match all nodes of a super element properly**. This is a common source for a later malfunction of the CAD converter Z88X !

5.6 PIPE UNDER INTERNAL PRESSURE, PLAIN STRESS ELEMENT NO.7

Copy the example file B6_X.DXF to Z88X.DXF.
B6_X.DXF---> Z88X.DXF input file for CAD converter Z88X

CAD:

Import Z88X.DXF into your CAD program and look at it. Usually you would have designed this example in a CAD system and then exported it as Z88X.DXF.

Z88: (in reduced form, more detailed instructions cf. examples 5.1, 5.2 and 5.3)

Z88X, conversion, "from Z88X.DXF to Z88I*.TXT"
Z88P, looking at structure, structure file Z88I1.TXT
Z88F calculates deflections
Z88D calculates stresses
Z88E calculates nodal forces
Z88P, plot FE structure, now also deflected (FUX, FUY, FUZ per 100.)

We deal with a pipe under internal pressure of 1,000 bar (=100 N/mm*mm). Inside diameter of the pipe is 80 mm, outside diameter of the pipe is 160 mm. The length is 40 mm. If one chooses the supports cleverly, a quarter of the pipe is enough to reflect the problem.

Such structures are best suited for polar coordinates. The internal pressure of 1,000 bar corresponds to a force of 251,327 N which is loaded onto the inside quadrant. The 251,327 N have to be distributed onto the nodes 1,6,9,14,17,22,25,30 and 33 in accordance with the rules for boundary conditions (cf. chapter 3.4):

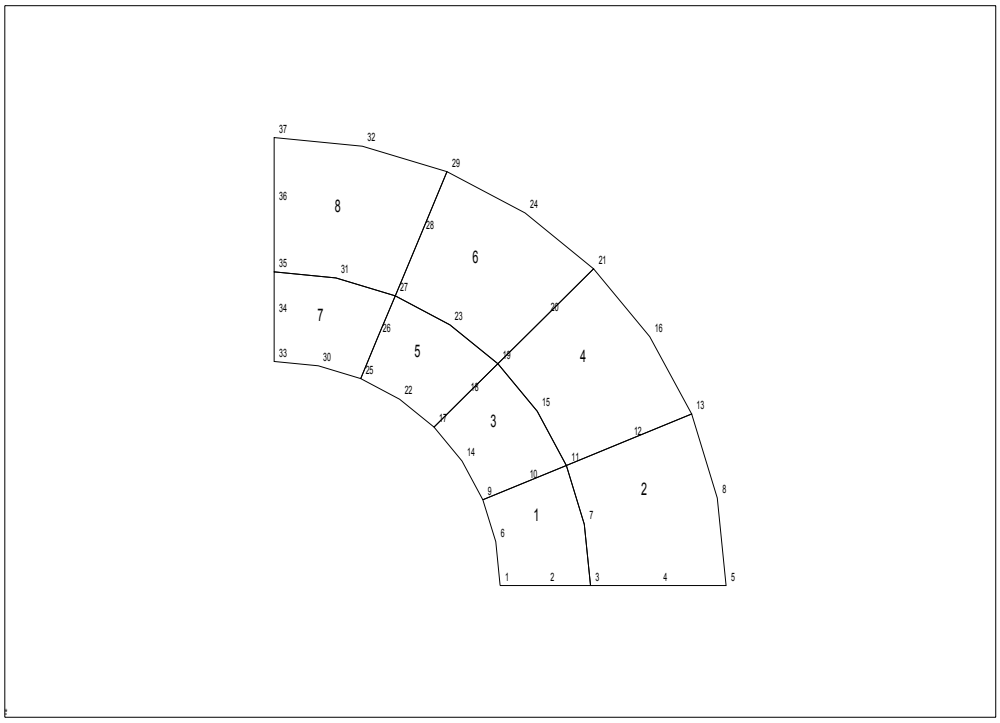
"1/6 points": 10,472 N
"2/3 points": 41,888 N
"2/6 points": 20,944 N

Control: 2*10,472+4*41,888+3*20,944 = 251,328 O.k.

These forces have an outwardly directed radial effect. Thus, they must be subdivided into X and Y components for boundary conditions. E.g. the node 6 as "2/3 point" is subdivided into X = 41,083 N and into Y = 8,172 N, because node 6 has an angle Phi = 11.25 degrees.

When dealing with a rotationally symmetrical structure, the additional calculation of radial stresses and tangential stresses can be interesting: Set KFLAG to 1 in Z88I3.TXT. As stresses are calculated in the Gauss points, use linear extrapolations to get the stresses directly in the inside diameter and the outside diameter.

This problem is simple to check analytically. Consult appropriate machine element books for proper calculation formulas or see chapter 5.7.



Plot of the undeflected structure

5.6.1 Input

With CAD program:

Proceed after the description chapter 2.7.2. Do not forget to write on the layer Z88EIO the element descriptions by TEXT function:

```
FE  1  7      (1st finite element type 7)
FE  2  7      (2nd finite element type 7)
.....         (element 3 to 7 dropped here)
FE  8  7      ( 8th finite element type 7)
```

Write the general information and material information on the layer Z88GEN, like

```
Z88I1.TXT  2  37  8  74  1  1  0  0   (2D, 37 nodes, 8 ele, 74 DOF, 1 mat info, polar
                                        coor., beam & plate flags 0, each)
MAT 1  1  8  206000  0.3  3  40       (1st mat info: Ele 1 to 8: Young's,  Poisson's,
                                       INTORD=  3, QPARA = thickness = 40)
```

Write the boundary conditions with the TEXT function onto the layer Z88RBD. Here we have the case of edge loads for the boundary conditions. You should consult chapter 3.4. and take into account the explanation and sketches for load distributions.

```
Z88I2.TXT  26              (26 boundary conditions)
RBD   1   1  1   1  10472  (1st BC: Node 1, DOF 1(= X), a load of 10,472 N)
RBD   2   1  2   2  0      (2nd BC: Node 1, DOF 2 (=Y), a displacement of 0 (=fixed))
RBD   3   2  2   2  0
RBD   4   3  2   2  0
RBD   5   4  2   2  0
RBD   6   5  2   2  0
```

```
RBD   7   6 1   1   41083
RBD   8   6 2   1   8172
RBD   9   9 1   1   19350
RBD  10   9 2   1   8015
RBD  11  14 1   1   34829
RBD  12  14 2   1   23272
RBD  13  17 1   1   14810
RBD  14  17 2   1   14810
RBD  15  22 1   1   23272
RBD  16  22 2   1   34829
RBD  17  25 1   1   8015
RBD  18  25 2   1   19350
RBD  19  30 1   1   8172
RBD  20  30 2   1   41083
RBD  21  33 1   2   0
RBD  22  33 2   1   10472
RBD  23  34 1   2   0
RBD  24  35 1   2   0
RBD  25  36 1   2   0
RBD  26  37 1   2   0
```

Switch to the layer Z88GEN and write into any free place:

```
Z88I3.TXT   3   1   1   (3x3 Gauss points for stresses, KFLAG=1 i.e. additional calculation
                          of radial and tangential stresses, von Mises stresses)
```

Export the drawing as DXF file with the name Z88X.DXF, then start the CAD converter Z88X with the option "from Z88X.DXF to Z88I*.TXT" (DXF -> I*). The CAD converter produces the three Z88 input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With editor:

Write the structure data file Z88I1.TXT (cf. chapter 3.2) with an editor:

```
2   37   8   74   1   1   0   0     ( 2D, 37 nodes, 8 elements, 74 DOF, 1 mat info, Polar coor.,
                                     beam & plate flags 0, each)
 1   2   40   0                     (1st node, 2 DOF, R and Phi coordinate)
 2   2   48   0                     (2nd node, 2 DOF, R and Phi coordinate)
 3   2   56   0
 4   2   68   0
 5   2   80   0
 6   2   40   11.25
 7   2   56   11.25
 8   2   80   11.25
 9   2   40   22.5
.....                                (Nodes 10.. 35 dropped here )
36   2   68   90
37   2   80   90
 1   7                               (element 1, Plain Stress Element No.7)
 1   3   11   9   2   7   10   6    (coincidence 1st element)
 2   7
```

```
  3   5   13  11  4   8   12   7
.....                              (elements 3 .. 7 dropped here)
  8   7                            (element 8, Plain Stress Element No.7)
27  29  37  35  28  32  36  31 (coincidence 8th element)
1  8  206000   0.3   3   40       (Ele 1 to 8: Young's, Poisson's, INTORD = 3, thickness = 40)
```

Here we have the case of edge loads for the boundary conditions. Consult chapter 3.4. and take into account the explanation and sketches for load distributions. Here is Z88I2.TXT:

```
 26                    (26 boundary conditions)
  1  1   1   10472     (1st BC: Node 1, DOF 1(= X), a load of 10,472 N)
  1  2   2   0         (2nd BC: Node 1, DOF 2 (=Y), a displacement of 0 (=fixed))
  2  2   2   0
  3  2   2   0
  4  2   2   0
  5  2   2   0
  6  1   1   41083
  6  2   1   8172
  9  1   1   19350
  9  2   1   8015
 14  1   1   34829
 14  2   1   23272
 17  1   1   14810
 17  2   1   14810
 22  1   1   23272
 22  2   1   34829
 25  1   1   8015
 25  2   1   19350
 30  1   1   8172
 30  2   1   41083
 33  1   2   0
 33  2   1   10472
 34  1   2   0
 35  1   2   0
 36  1   2   0
 37  1   2   0
```

This example is very nice for experiments with the boundary conditions: Enter deflections rather than forces into X and Y, e.g. 0.01 mm in radial direction to the outside. At node 1 you can enter the 0.01 mm directly as X displacement and at node 33 you can enter directly the Y displacement of 0.01 mm, but for the other nodes the radial displacements of 0.01 mm must be subdivided into X and Y components respectively (via sine and cosine). Or enter mixed BC: A couple of nodes with displacements, the others with forces.. In practice nobody would do so for such a task, however, but Z88 can handle it.

A broad experimenting field also opens up with Z88I3.TXT: You have 5 possibilities for the first value and two possibilities each for the second and third value, cf. Chapters 3.5 and 4.7.

Now we can produce plenty of results:

Here is Z88I3.TXT:

3 1 1 *(3x3 Gauss points for stresses, KFLAG=1 i.e. additional calculation of radial and tangential stresses, von Mises stresses)*

CAD and editor:

The structure data Z88I1.TXT, the boundary conditions Z88I2.TXT and the header file for the stress processor Z88I3.TXT (with any contents) are ready to go. Now launch

> Z88F the Cholesky solver
> Z88D the stress processor
> Z88E the nodal force processor

5.6.2 Results:

The Cholesky solver Z88F provides the following output files:

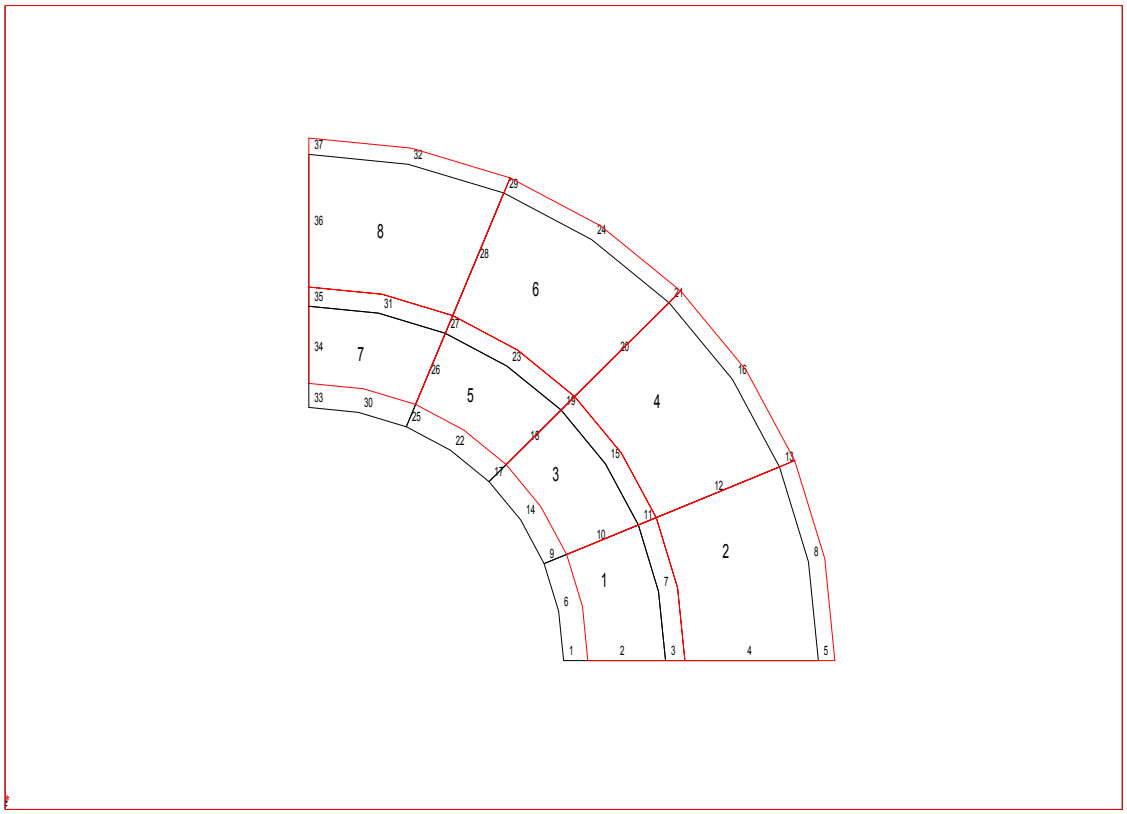
Z88O0.TXT stores the processed structure data. For documentation purposes.
Z88O1.TXT stores the processed boundary conditions: For documentation purposes.
Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

The stress processor **Z88D** internally uses the calculated displacements from Z88F and stores **Z88O3.TXT**, the calculated stresses.

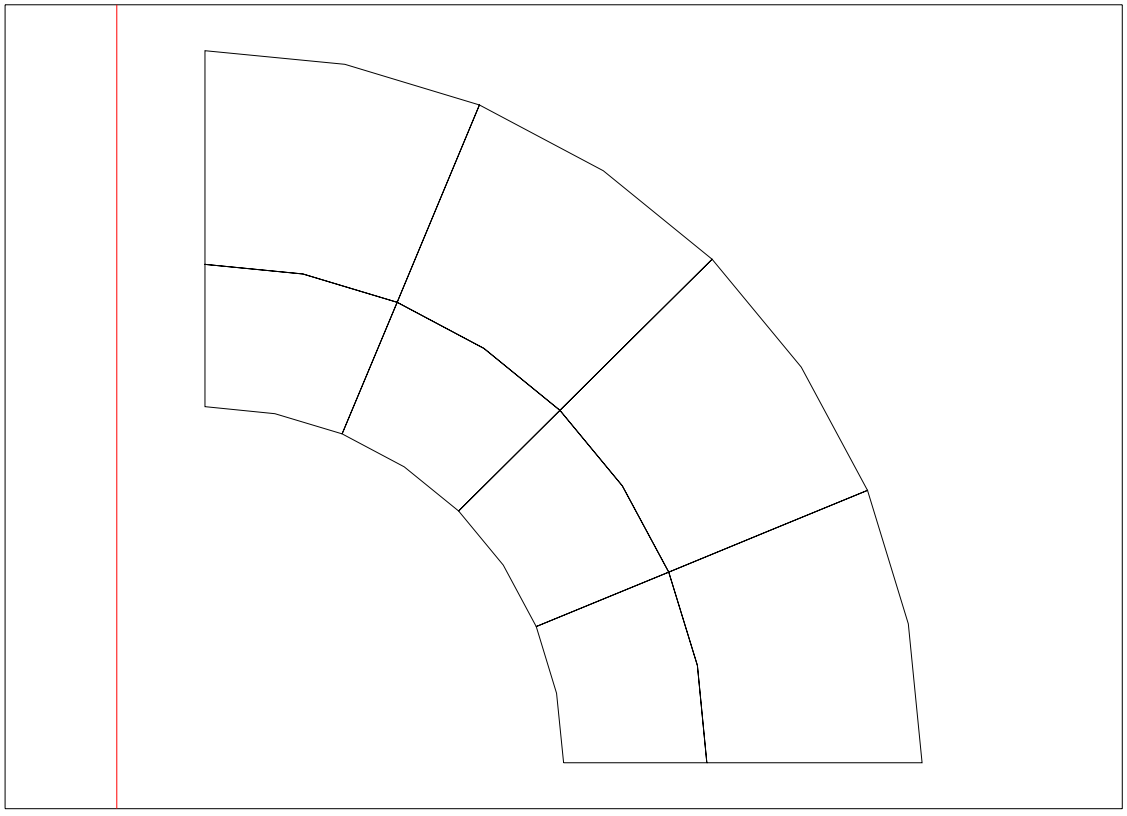
The nodal force processor **Z88E** internally uses the calculated deflections of Z88F and stores **Z88O4.TXT**, the computed nodal forces.

The following picture of the plot program shows the deflected structure for FUXand FUY = 100 each (magnifications of the deflections):

This example is very suitable to demonstrate all the possibilities of the stress calculation with Z88D and Plain Stress Elements No.7 (or Plain Stress Elements No.11). We recall: Z88I3.TXT was: 3 1 1, i.e. 3 x 3 Gauss points, additional calculation of radial and tangential stresses (which is very meaningful for this example) and von Mises stresses calculation. Enter in Z88I3.TXT: 3 0 1, so that you will get von Mises stresses but no radial and tangential stresses. The results with 2 0 0 become even shorter (only still 2 x 2 Gauss points, no radial/tangential stresses and no von Mises stresses). You will get the stresses with 0 0 0 into the corner nodes instead of into the Gauss points. Take into account that stresses display in Z88P is impossible for corner nodes. Now experiment.. you have 5 x 2 x 2 = 20 possibilities.



Plot of the undeflected and the deflected structure



Plot of the *von Mises* stresses

5.7 PIPE UNDER INTERNAL PRESSURE, TORUS NO.8

Copy the example files B7_* to Z88 entry files Z88*:

B7_X.DXF --> Z88X.DXF input file for the CAD converter Z88X
B7_2.TXT --> Z88I2.TXT boundary conditions
B7_3.TXT --> Z88I3.TXT heading parameter for tension processor

CAD:

Import Z88X.DXF into your CAD program and view the superstructure. You usually would have designed this example in a CAD system and then exported it as Z88X.DXF.

Z88:

Z88X, conversion "from Z88X.DXF to Z88NI.TXT"
Z88P, structure file Z88NI, look at the super structure
Z88N, mesh generator, produces Z88I1.TXT
Z88P, structure file Z88I1.TXT, undflected FE structure
Z88X, conversion, "from Z88I* . TXT to Z88X.DXF"

CAD:

Import Z88X.DXF into your CAD program and look at it. You usually would have now added the boundary conditions and control information Z88I3.TXT into CAD and then exported it as Z88X.DXF.

Z88:

Z88X, conversion, "from Z88X.DXF to Z88I* . TXT"
Z88F calculates deflections
Z88D calculates stresses
Z88P, plots FE structure, now also delected and stresses display
Z88E, nodal force calculation

We look at a pipe under internal pressure. Pipe inside diameter 80 mm, pipe outside diameter 160 mm, length 40 mm. For Torus elements the cross-section of the pipe is important.

The inside radius shall be expanded by 0.1 mm = rd (press fit). Attach this displacements to the nodes from 1 to 11. To fix the structure in space, e.g. fix node 6 in Z direction.

One calculates analytically:

p = rd*E/ri*(1/((1+qa)/(1-qa) +nue)) = 262 N/mm**2 = 2.620 bar
with qa= ri**2/ra**2 = 0.25 and E = Young's modulus and nue = Poisson's ratio

Radial stresses:	SIGRR i	= -p	= -262 N/mm*2
	SIGRR a	= 0	= 0
Tangential stresses:	SIGTE i	= p*((1+qa)/(1-qa))	= 437 N/mm**2
	SIGTE a	= 2p*qa/(1-qa)	= 175 N/mm**2

Because stresses are calculated in the Gauss points, use linear extrapolations to get the

stresses directly in the inside diameter and the outside diameter.

The force: F = p*A = p*2*Pi*ri*l = 2,633,911 N.
This confirms the sum of the forces of the elements 1-5 for the nodes 1-11 in Z88O4.TXT.

5.7.1 Input

General: The entries for the mesh generator contain merely a single Torus No.8 as super element. It is subdivided into 40 finite elements. A Torus No.12 also could, of course, be used as super element. Yet this is quite useless for this simple super structure, being designed of straight lines. Torus elements No.12 are more powerful than Torus elements No.8 if the super structure has many curvilinear edges because they feature cubic shape functions, but Torus No.8 uses only square parables. Thus, many curvilinear structures allow a better approach with few Torus elements No.12 due to the higher curve function.

Make sure that cylindrical coordinates are always expected for Torus No.6, No.8 and No.12, i.e. radius R (replaces X) and height coordinate Z (replaces Y). R and Z must feature always positive values ! KFLAG must be zero!

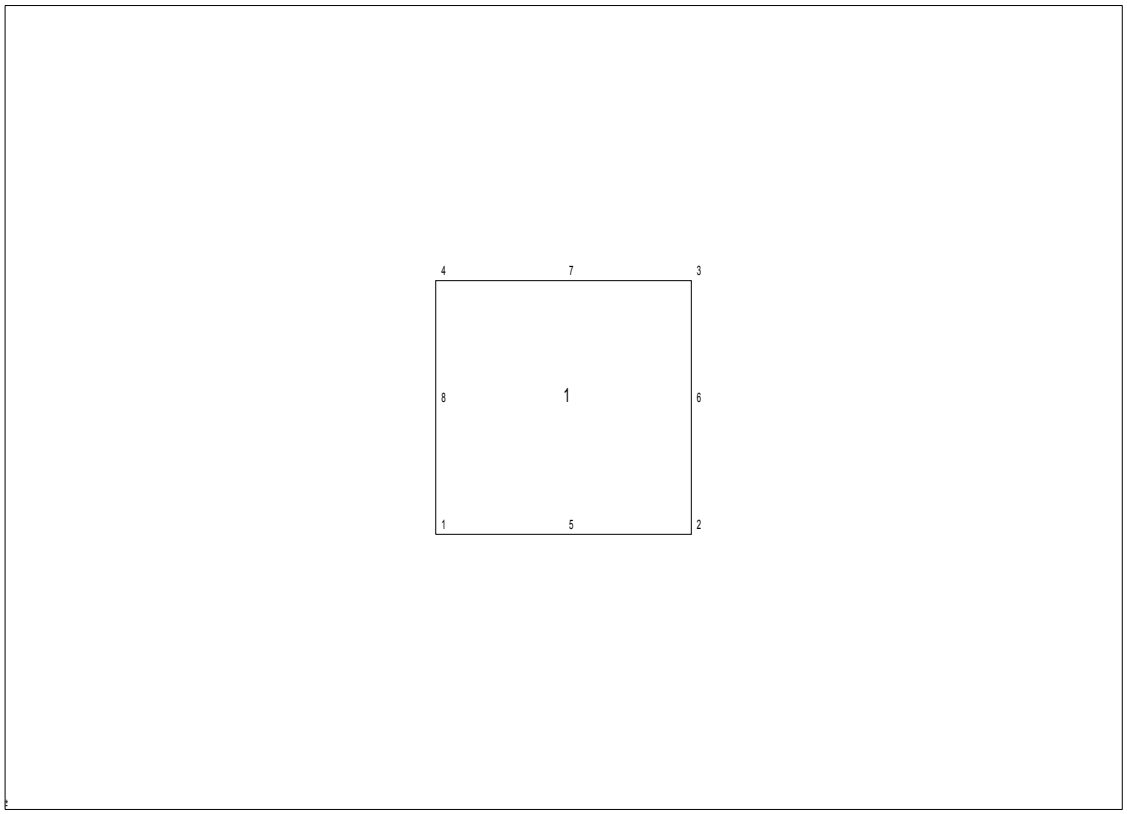
With CAD program:

Proceed after the description chapter 2.7.2. Do not forget to write on the layer Z88EIO the super element descriptions by TEXT function:

SE 1 8 8 L 5 e *(subdivide 8x into X geometrical ascending and 5x equidistant into Y)*
Write the general information and material information on the layer Z88GEN,

Z88NI.TXT 2 8 1 16 1 0 0 0 0 *(2D, 8 nodes, 1 SE, 16 DOF, 1 mat info, all flags 0)*
MAT 1 1 1 206000 0.3 3 0 *(SE1 to SE1:Young's,Poisson's,INTORD for FE, QPARA=0)*

Export the drawing as DXF file with the name Z88X.DXF and start the CAD converter Z88X with the option "from Z88X.DXF to Z88NI.TXT". Z88X will produce the mesh generator input file Z88NI.TXT. (You should have a look at it with Z88P).



With editor:

Write the mesh generator input file Z88NI.TXT (cf. chapter 3.3) with an editor:

```
2  8  1  16  1  0  0  0  0  (2D, 8 nodes, 1 SE, 16 DOF, 1 mat info, all flags 0)
1   2   40    0             (1st node,  2 DOF, R and Z coordinate)
2   2   80    0             (2nd node, 2 DOF, R and Z coordinate)
3   2   80   40
4   2   40   40
5   2   60    0
6   2   80   20
7   2   60   40
8   2   40   20
1   8                       (superelement 1, type Torus No.8)
1  2  3  4  5  6  7  8       (coincidence 1st SE)
1  1  206000  0.3  3  0     (SE1 to SE1: Young's,Poisson's,INTORD for FE,QPARA=0)
1   8                       (subdivide SE1 into Torus elements No.8 and subdivide)
8   L   5   E               (8 times geometrical ascending into X and 5 times equidistant into Y)
```

CAD and editor:

Start the mesh generator Z88N to produce the final Z88 structure file Z88I1.TXT. Look at it either

- in the CAD program (from Z88I1.TXT to Z88X.DXF) after conversion with Z88X or
- with the Z88 plot program Z88P for defining the boundary conditions:

We force displacements of 0.1 mm upon the inside margin. Every node receives the same value as the load division in accordance with section 2.4 applies to forces only. Take care that the structure is fixed in space again. Therefore fix the degree of freedom 2 for the node 6. Any other nodes are possible, too.

With CAD program:

Switch to the layer Z88RBD and write with the TEXT function into any free place:

```
Z88I2.TXT   12          (12 boundary conditions)
RBD   1   1   1   2   0.1    (RB 1: node 1, at DOF 1, i.e into R, a displacement of 0.1 mm)
RBD   2   2   1   2   0.1
RBD   3   3   1   2   0.1
RBD   4   4   1   2   0.1
RBD   5   5   1   2   0.1
RBD   5   6   1   2   0.1
RBD   7   6   2   2   0      (BC 7: for fixing structure in space)
RBD   8   7   1   2   0.1
RBD   9   8   1   2   0.1
RBD  10   9   1   2   0.1
RBD  11  10   1   2   0.1
RBD  12  11   1   2   0.1
```

With editor:

Create the file of the boundary conditions Z88I2.TXT by editing:

```
12                  (12 boundary conditions)
 1   1   2   0.1    (RB 1: node 1, at DOF 1, i.e into R, a displacement of 0.1 mm)
 2   1   2   0.1
 3   1   2   0.1
 4   1   2   0.1
 5   1   2   0.1
 6   1   2   0.1
 6   2   2   0      (BC 7: for fixing structure in space)
 7   1   2   0.1
 8   1   2   0.1
 9   1   2   0.1
10   1   2   0.1
11   1   2   0.1
```

Input for stress calculation:

In the CAD program:

Switch to the layer Z88GEN and write with the TEXT function into any free place:

```
Z88I3.TXT   3   0   1   (3 x 3 Gauss points per FE, KFLAG 0, von Mises stresses)
```

KFLAG always 0, because additional output of radial and tangential stresses is useless for torus elements. SIGRR (radial stresses) and SIGTE (tangential stresses) are calculated for torus elements anyway, cf. section 4.12.

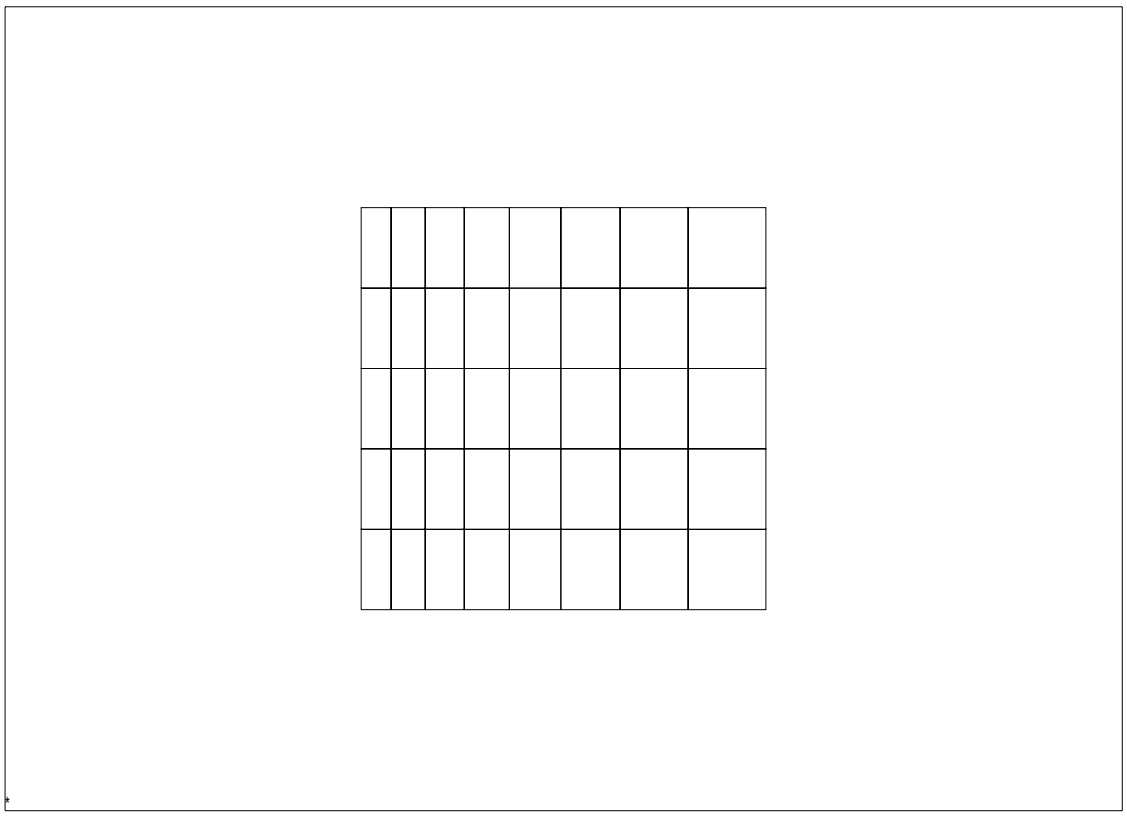
Export the drawing as DXF file with the name Z88X.DXF, then start the CAD converter

Z88X with the option "from Z88X.DXF to Z88I*.TXT". The CAD converter produces the three Z88 input files Z88I1.TXT, Z88I2.TXT, Z88I3.TXT.

With editor:

Enter in the parameter file for the stress processor Z88I3.TXT (cf. Chapter 3.5):

3 0 1 *(3x3 Gauss points for stresses, KFLAG 0, von Mises stresses)*



FE mesh Z88I1.TXT

CAD and editor:

Now launch the Cholesky solver Z88F and then the stress processor Z88D. You should choose the Compactmode for Z88F because only one set of boundary conditions is available, cf. section 2.1. Compute nodal forces with Z88E.

5.7.2 Results

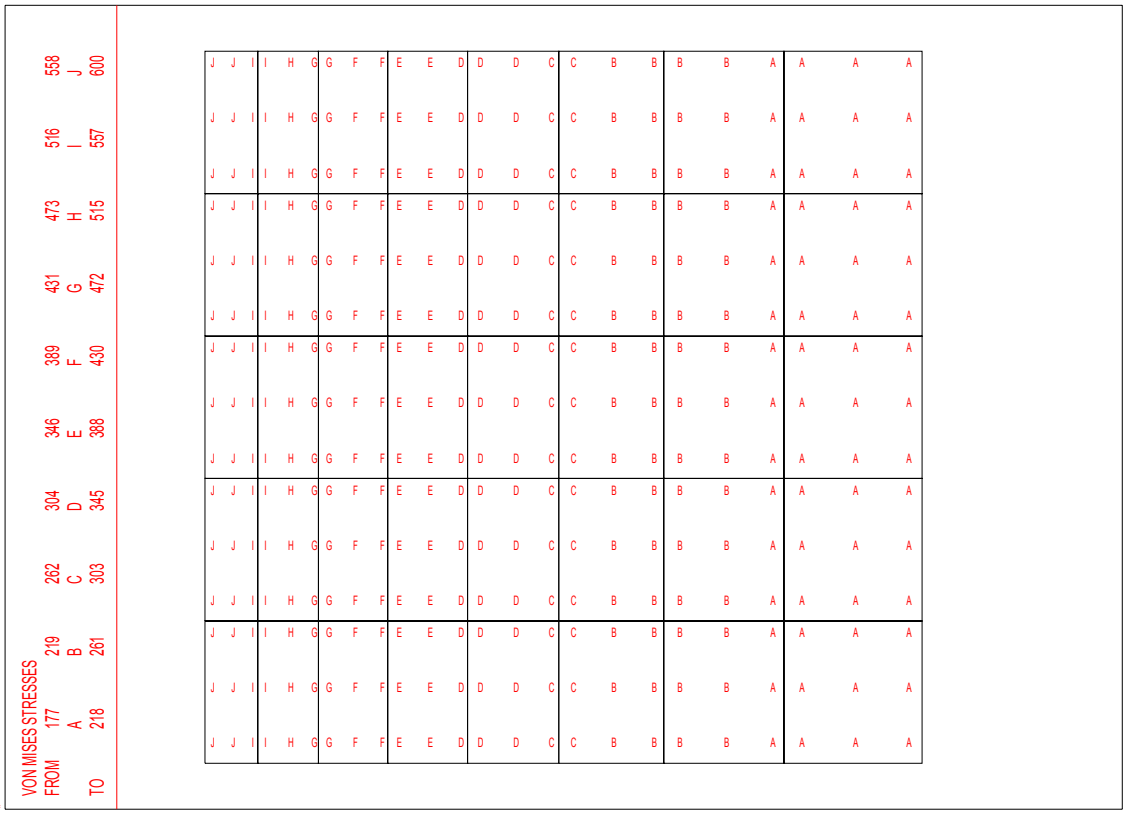
The Cholesky solver Z88F provides the following output files:

Z88O0.TXT stores the processed structure data. For documentation purposes.
Z88O1.TXT stores the processed boundary conditions: For documentation purposes.
Z88O2.TXT, the displacements, the main task and solution of the FEA problem.

The stress processor **Z88D** uses internally the calculated displacements from Z88F and stores **Z88O3.TXT**, the calculated stresses. The results in Z88O3.TXT depend on the header parameters in Z88I3.TXT.

The nodal force processor **Z88E** uses internally the calculated deflections of Z88F and stores

Z88O4.TXT, the computed nodal forces.



Stresses display of the torus structure

5.8 MOTORCYCLE CRANKSHAFT, TETRAHEDRON NO. 16

Copy the sample file B11_G.COS to the Z88 input file Z88G.COS.

We want to compute a crankshaft for a monocylinder motorcycle engine and put a force of -5,000 N onto the piston. The meshing will do Pro/ENGINEER.

The boundary conditions are a bit tricky for this example: Put a reference (or datum) point to the center of the face of the crankshaft. We'll need this point to fix the crankshaft in Z direction, i.e. lengthwise.

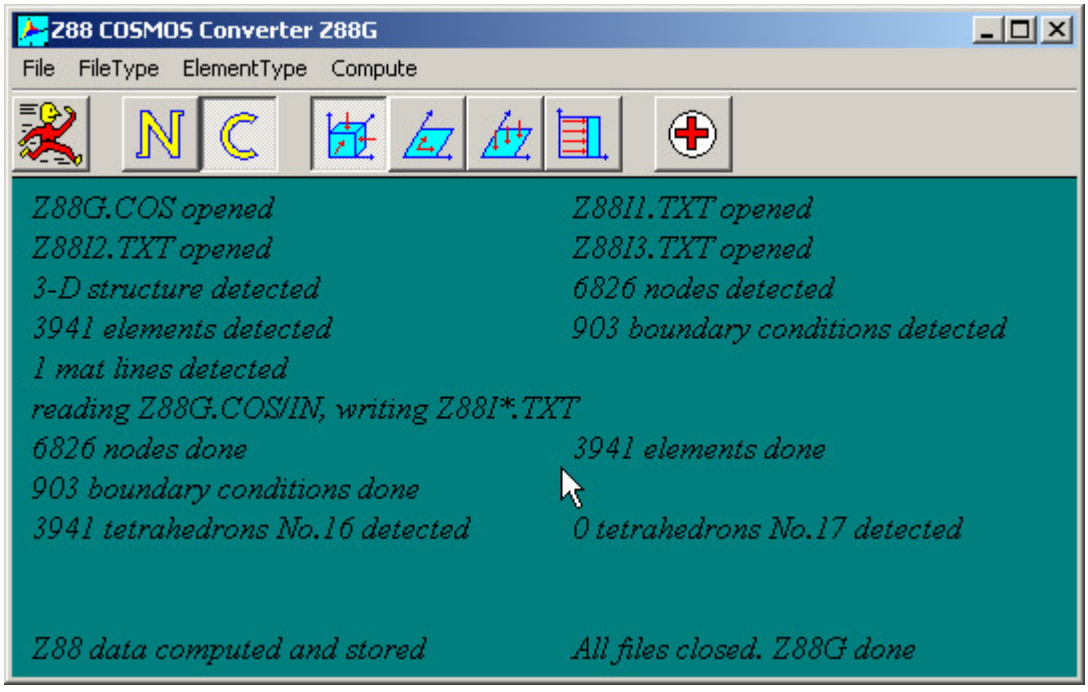
The ball bearings, which allow always some angular movement, and, thus, should be regarded as moment- free supports, are fastened to the larger shaft axises. The flange facings of the shaft axises are to be fixed in X and Y direction. Because whole surfaces are fixed, don't allow one ore more of these surfaces to be fixed in Z direction, too. This would result in blocking the angular movement - try it, if you won't believe it.

A total force of -5,000 N will be put onto the peripheral surface of the crankshaft journal.

The mesh is automatically generated by Pro/MECHANICA featuring parabolic tetrahedrons. After storing the COSMOS file, a Z88 session may start:

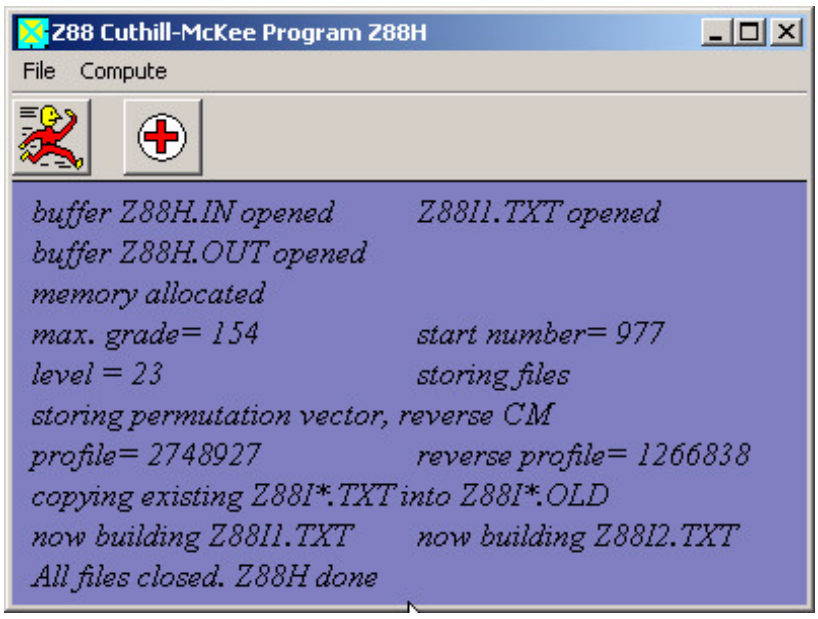
Copy B11_G.COS to Z88G.COS, the COSMOS file for the converter Z88G

Start converting Z88G.COS with **Z88G**



(Windows: COSMOS converter Z88G. Looks quite similar on UNIX machines)

and proceed with the Cuthill- McKee algorithm **Z88H**, because we'll expect a very bad node-numbering for the parabolic tetrahedrons.



(Windows: Cuthill- McKee program Z88H. Looks quite similar on UNIX machines)

The first line of Z88I1.TXT tells you the following values:

- 6,826 nodes
- 3,941 elemente
- 20,478 dof

MAXKOI must have as a minimum 3,941 elements * 10 nodes per element = 39,410.

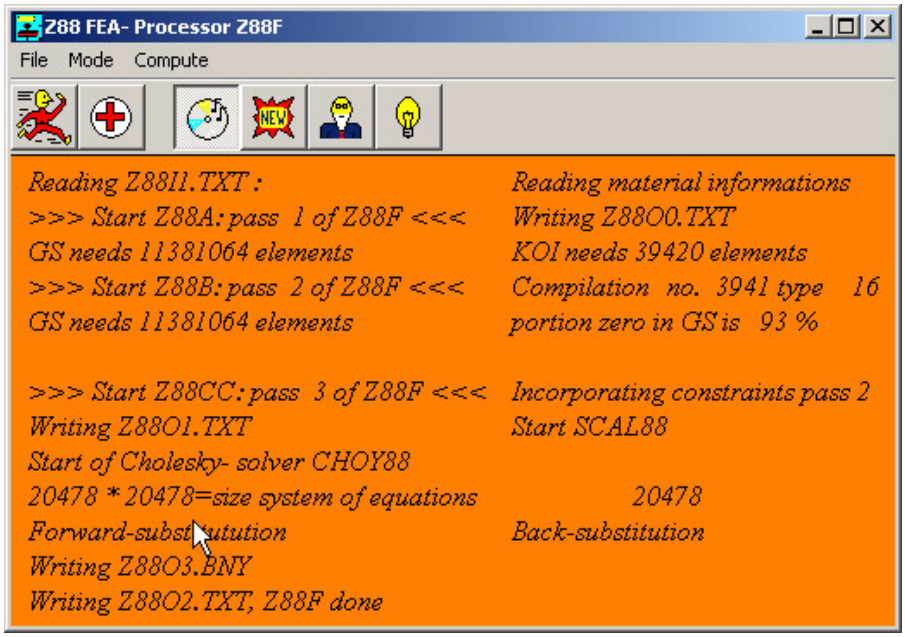
Thus, **Z88.DYN** should look as follows:

MAXGS when starting, any value
MAXKOI minimum 39410
MAXK minimum 6826
MAXE minimum 3941
MAXNFG minimum 20478
MAXNEG minimum 1

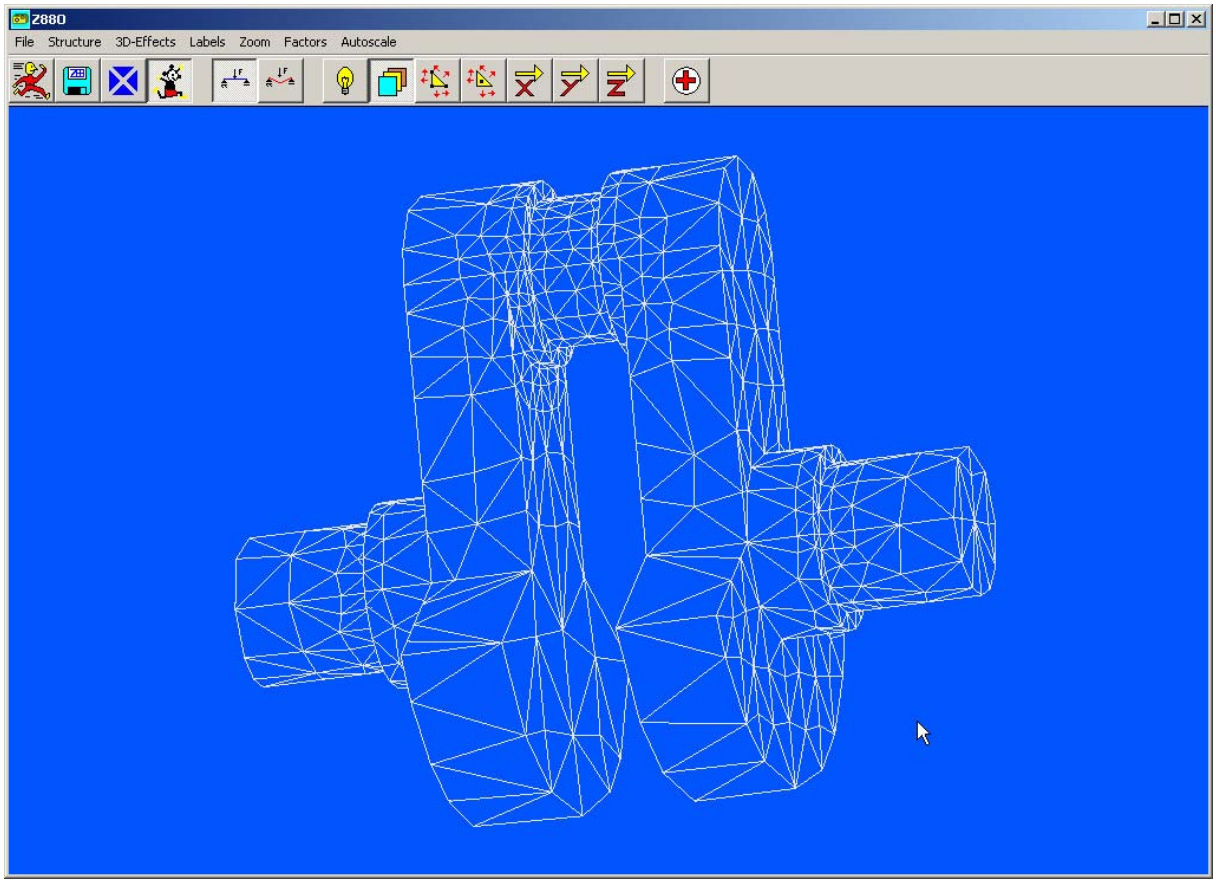
Proceed with a look at the structure with **Z88O** (or with Z88P).

The computing time with **Z88F** is about 1,5 minutes on a PC (900 MHz AMD- processor, 512 MByte memory). Enter a value of about 11,400,000 for MAXGS..

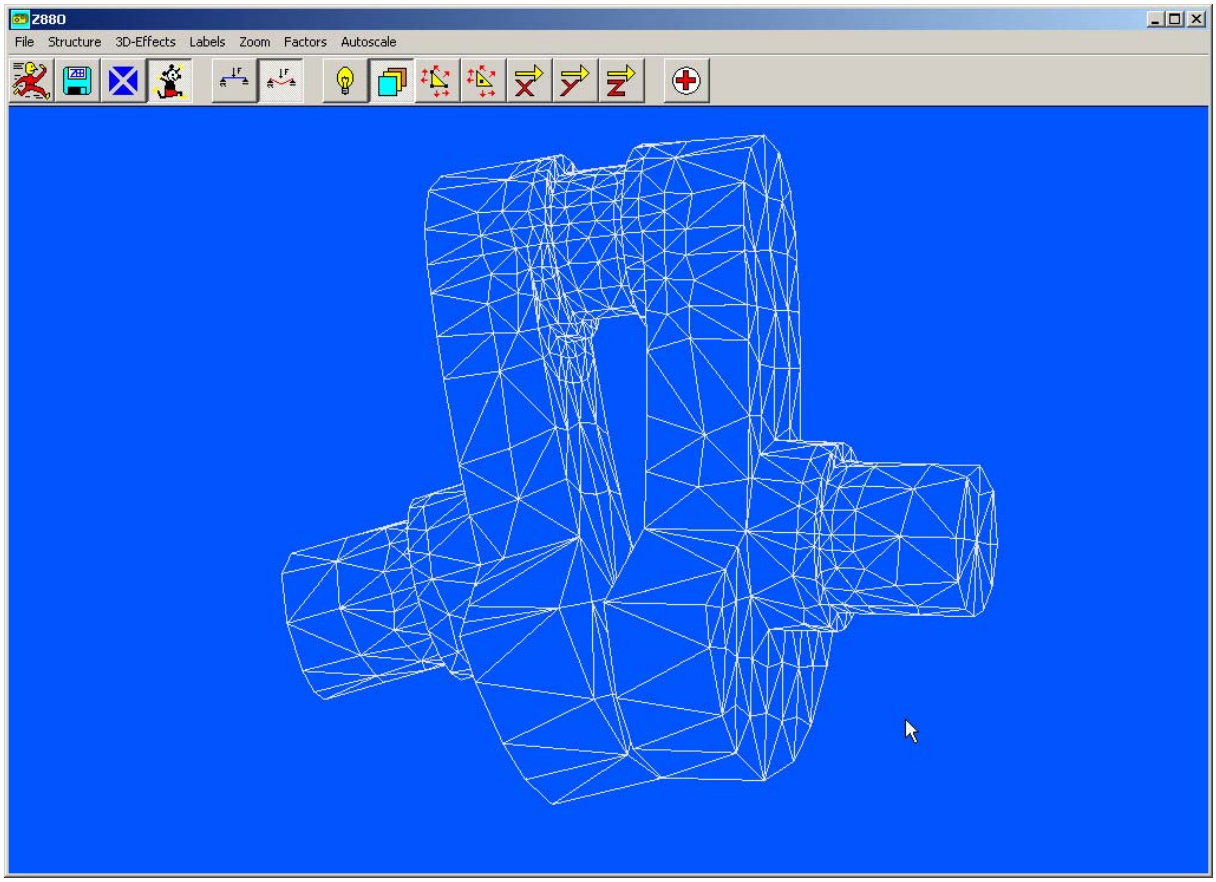
See the deflected structure with **Z88O**. The angular deflection of the axises is quite amazing. Now you would read off the deflections of distinguished nodes, multiply with the appropriate lever arms and check with the bearing catalogue if your ball bearings will allow this angular movement without problems.



(Windows: Computing deflections with Z88F. Looks quite similar on UNIX machines)



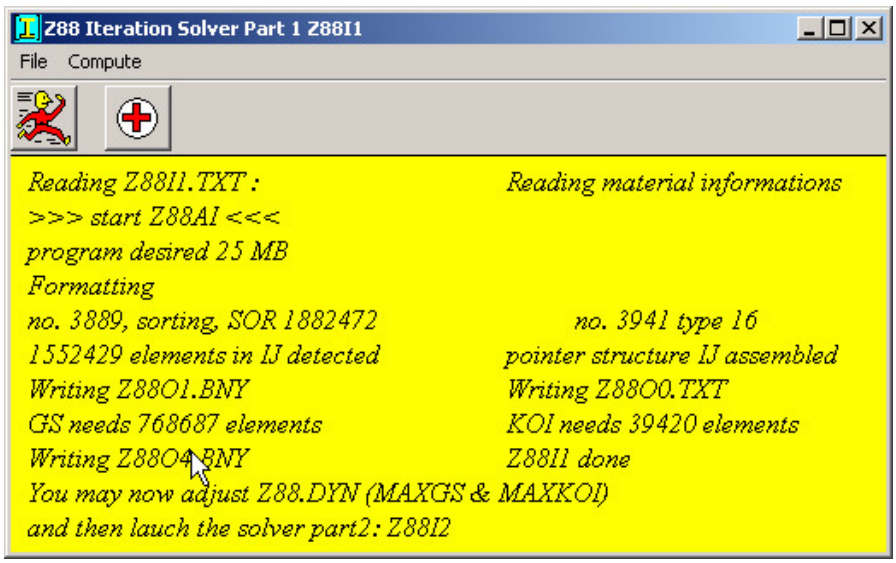
(Windows: Plot programm Z88O, undeflected structure)



(Windows: Plot programm Z88O, deflected structure)

Now we'll launch the iteration solver Z88I1 and Z88I2. To begin with, we'll try some values for MAXSOR and MAXPUF in Z88.DYN (you may also enter, for example, 50,000,000 for MAXSOR and 5,000,000 for MAXPUF, if you want):

```
COMMON START
    MAXGS  11500000     ← has for Z88I1 no meaning !
    MAXKOI    40000     ← must always be large enough !
    MAXK       7000     ← read off from Z88I1.TXT
    MAXE       4000     ← read off from Z88I1.TXT
    MAXNFG    21000     ← read off from Z88I1.TXT
    MAXNEG        1     ← read off from Z88I1.TXT
    MAXSOR  2000000     ← important for Z88I1
    MAXPUF   500000     ← important for Z88I1
COMMON END
```



Our entries did work properly (otherwise, you would have to increase MAXSOR and MAXPUF) and the sorting times was about 15 seconds on a PC (900 MHz AMD- processor, 512 MByte memory).

Read off for MAXGS: 768,687, rounded up 770,000. This looks fairly better than the direct Cholesky solver Z88F with its need of 11,381,064 8- Byte elements = 87 Mbyte. The second part of the iteration solver, i.e. Z88I2, will only need 768,687 8- Bytes elements = 6 MByte.

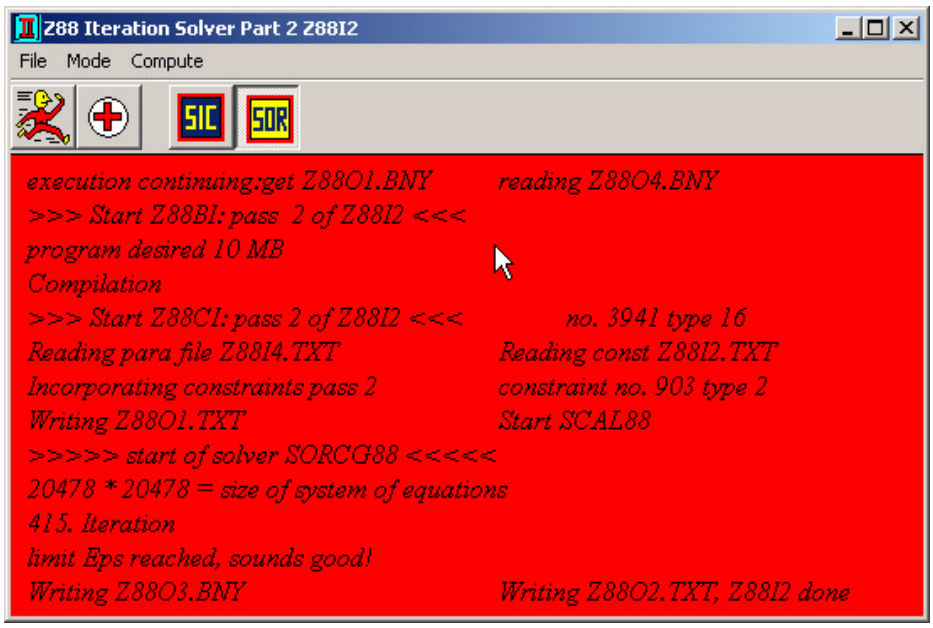
Thus, we would adjust the memory in Z88.DYN as follows (feel free to enter even bigger values):

```
COMMON START
    MAXGS     770000     ← important !
    MAXKOI     40000     ← must always be large enough !
    MAXK        7000     ← read off from Z88I1.TXT
    MAXE        4000     ← read off from Z88I1.TXT
    MAXNFG     21000     ← read off from Z88I1.TXT
    MAXNEG         1     ← read off from Z88I1.TXT
    MAXSOR   2000000     ← not used by Z88I2
    MAXPUF    500000     ← not used by Z88I2
COMMON END
```

If you adjust the iteration parameters in Z88I4.TXT (chapter 3.6) as follows:

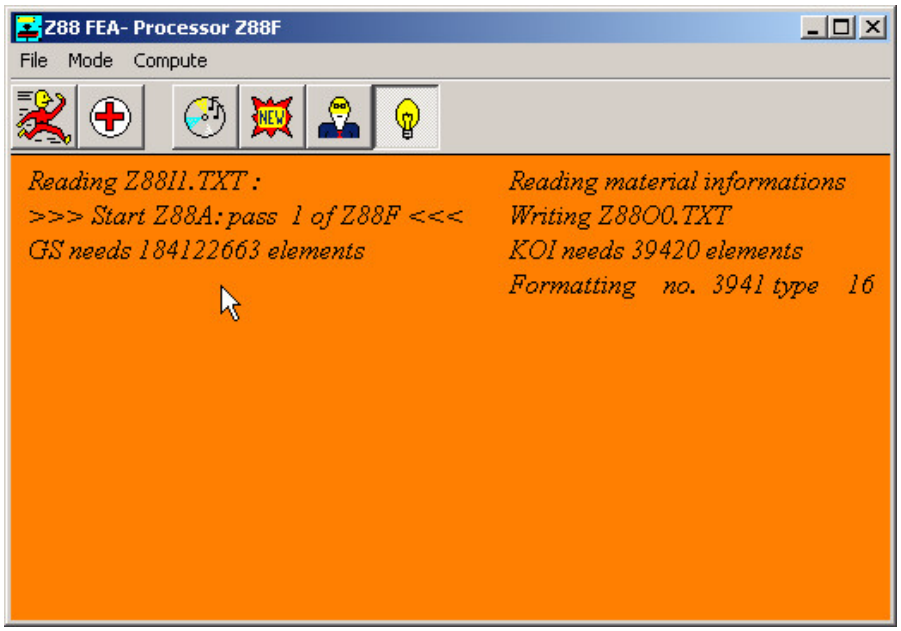
```
10000   1e-7  1.
```

i.e. a maximum of 10,000 iterations, *EPS* with 1E-7 and *RP* (here *Omega*) with 1, then this results in a computing time of about 1 minute on a PC (900 MHz AMD- processor, 512 MByte memory).
In this case, both the iteration solver and the direct Cholesky solver need about the same time, but the iteration solver needs fewer than one tenth of memory. For large structures, things get even worse for the Cholesky solver! But pay attention to the fact, that you can't really compare the computing times. Try other entries for *EPS*, for example 1E-5 (resulting in 303 iterations and 45 seconds) or 1E-10 (resulting in 474 iterations and 1:08 minutes), and see the different computing times.



(Windows: The iteration solver Part 2, i.e. Z88I2)

However, a very nice experiment is this:
Start from the very beginning, run Z88G, but not the Cuthill- McKee algorithm Z88H. Launch directly after Z88G a test run with Z88F (UNIX: z88f -t):



(Windows: The direct Cholesky solver in test mode)

Gee, see the faces falling: now we would need 184,122,663 8- Byte elements = 1,4 GByte. Absolutely no need for this!

However, run again the iteration solver part 1, i.e. Z88I1. This will again result in only 768,687 elements for the total stiffness matrix. Calculate, please:

184,122,663 : 768,687 = 240 : 1

The second part of the iteration solver, i.e. Z88I2, needs now some more iterations (451 in contrary to 415 with an equal *EPS* of 1E-7), because the matrix features the same number of

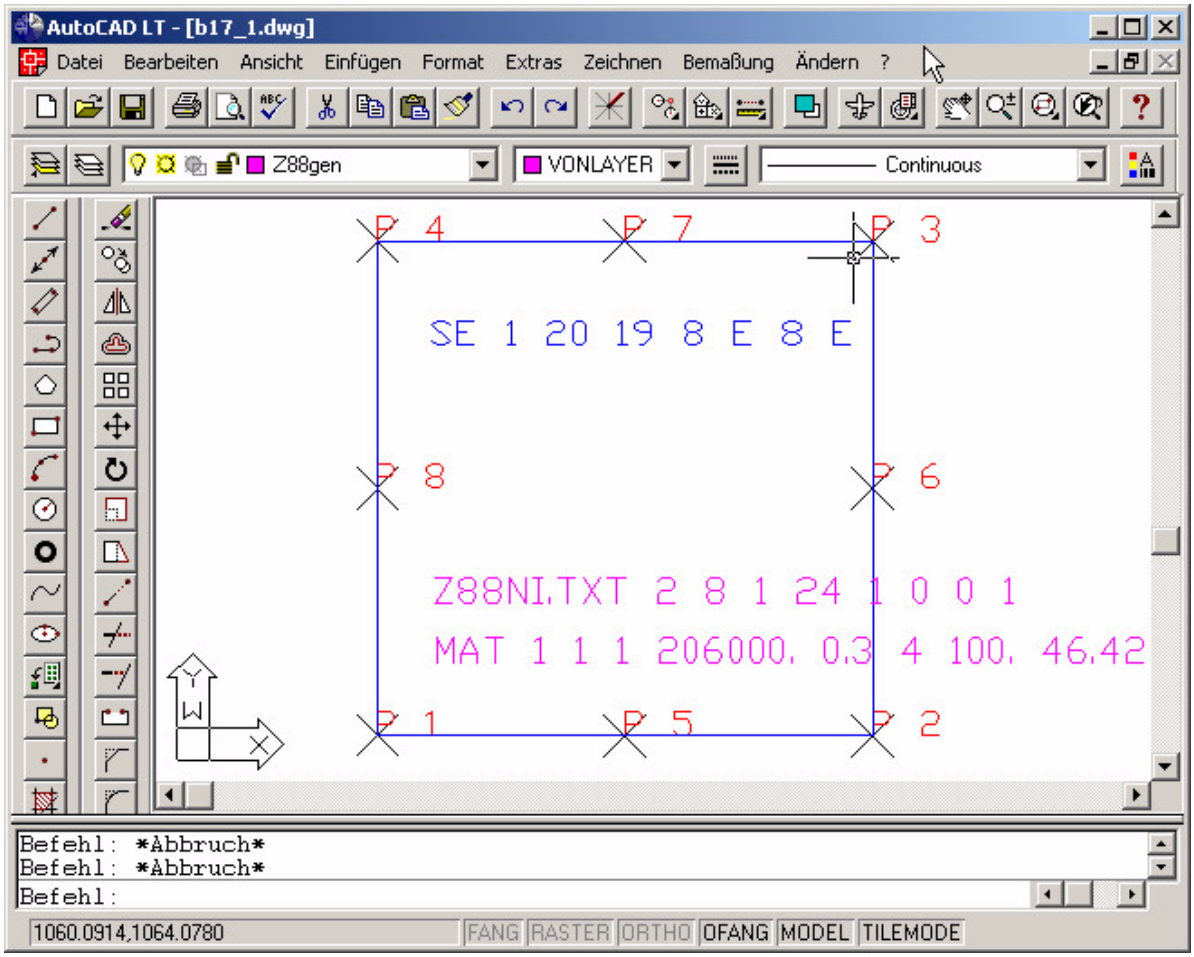
non- zero elements, though, but the condition is worse because of the very bad node-numbering of Pro/MECHANICA. That means: When using the iteration solver you don't need to run the Cuthill- McKee algorithm Z88H for reducing the storage needs of the iteration solver (in contrary to the direct Cholesky solver Z88F, which may depend heavily on Z88H for larger structures!). However, Z88H may improve the matrix condition anyway.

5.9 RECTANGULAR PLATE, PLATE NO. 19

We want to compute a thick rectangular plate of steel.
Data:

- Dimensions: 1.000 x 1.000 x 100 mm
- Area load 46,42 N/mm**2
- Young's Modulus 206.000 N/mm**2
- Poisson's Ratio 0.3

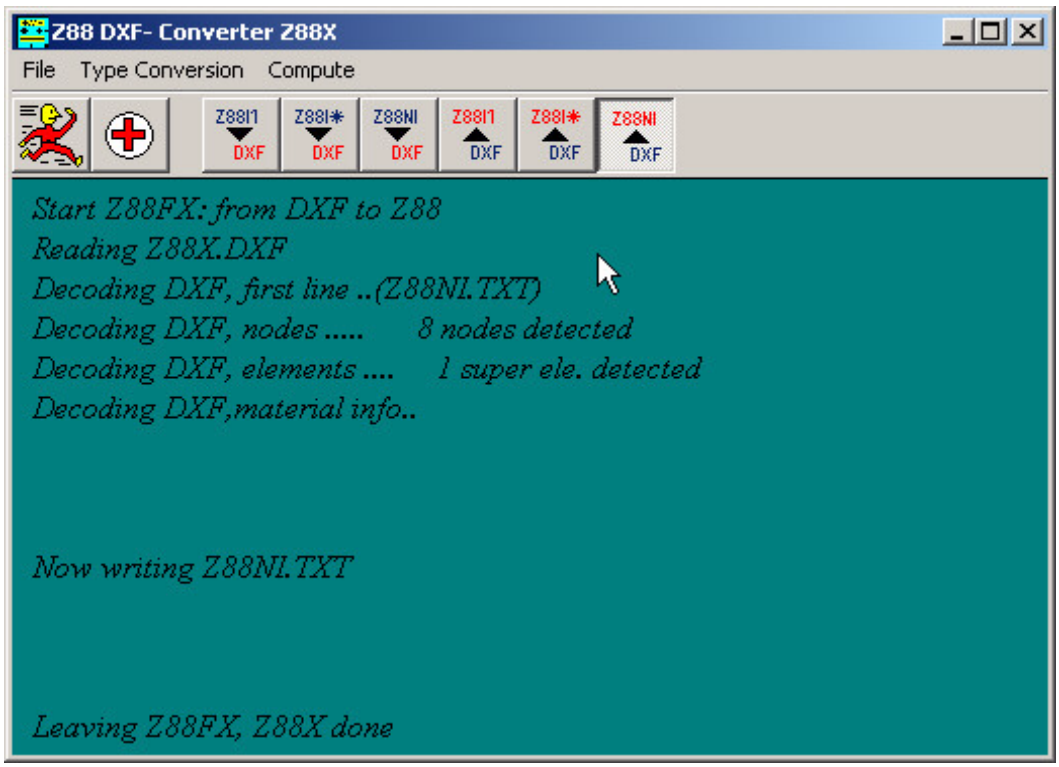
We will draw the plate super structure in AutoCAD. Draw one single super element plate type No.20, which will be subdivided by the mesher Z88N into 8 x 8 = 64 plates of type No.19, i.e. with 16 nodes each. Of course, for this example you could use an editor and generate the mesh generator input file by hand at the same pace:



(Windows: AutoCAD LT 97, drawing the rectangular plate)

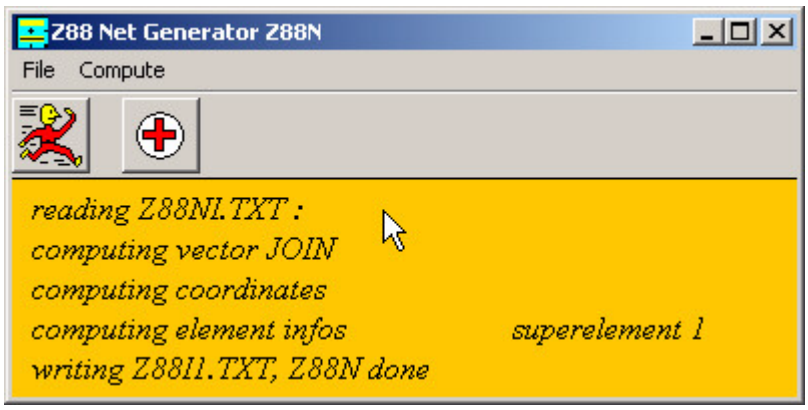
You'll find the exact procedure plottet in chapter 2.7 - however, try it by yourself and export the drawing as Z88X.DXF into the Z88 directory. If it doesn't work at all (but it really does):

Copy B18_X.DXF into Z88X.DXF



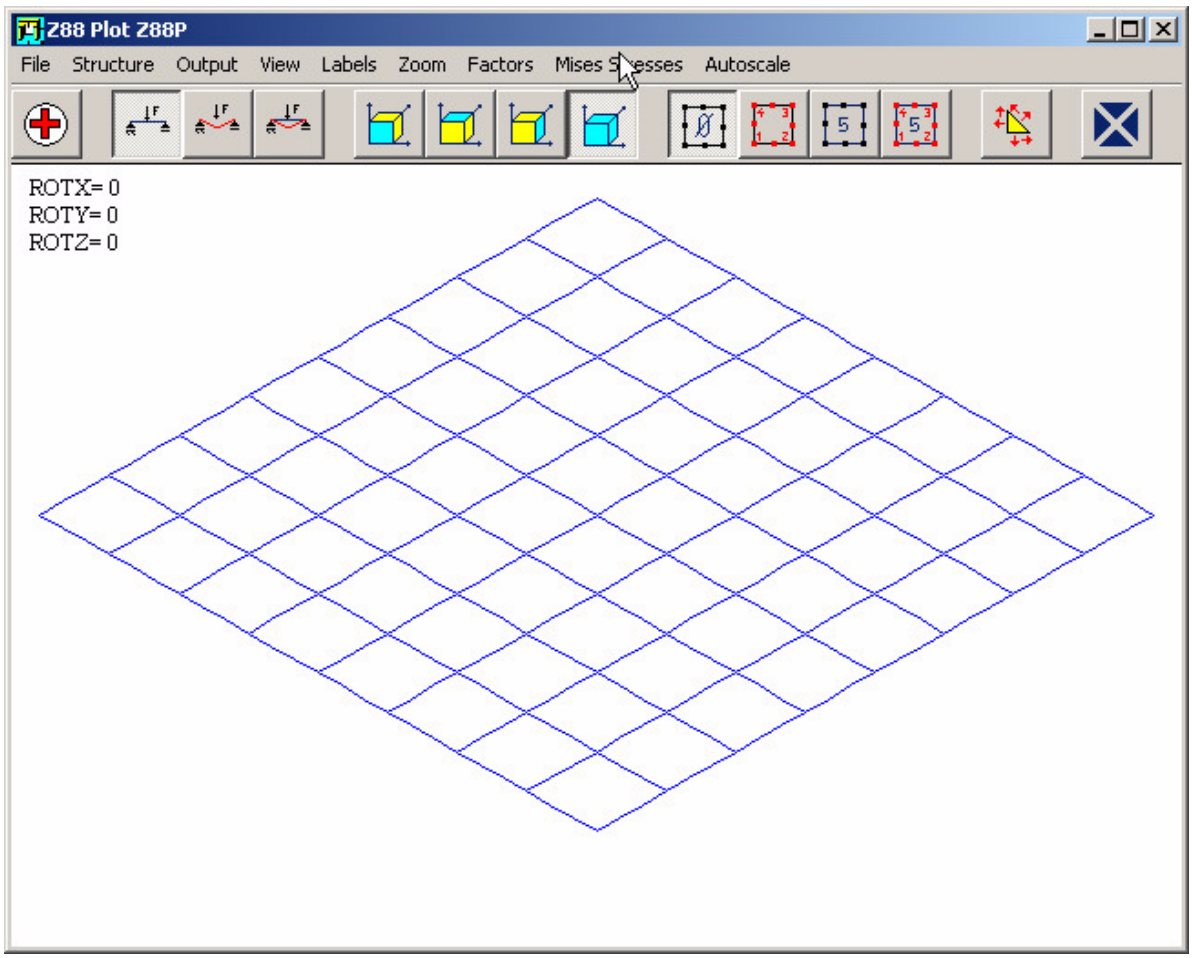
(Windows: CAD converter Z88X. Looks very similar on UNIX machines)

Choose *from Z88X to Z88NI.TXT*. Then, launch the mesher Z88N:



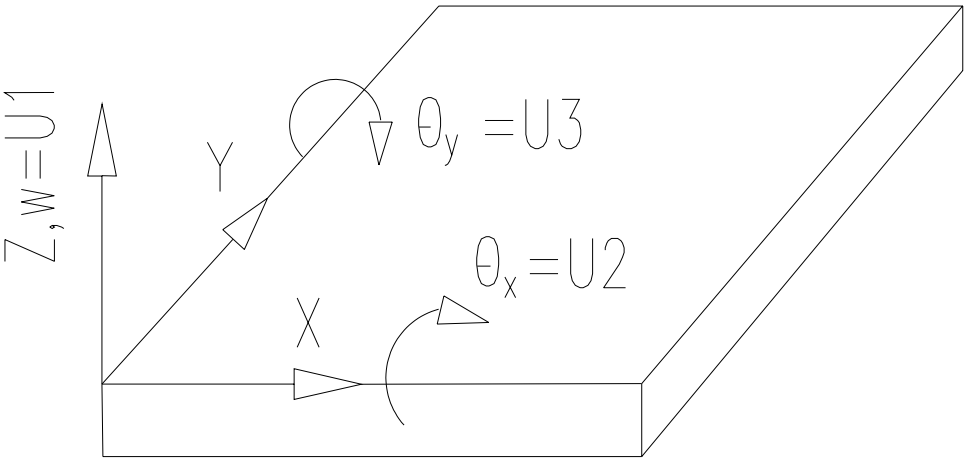
(Windows: mesh generator Z88N. Looks very similar on UNIX machines)

Now you may look at the structure with **Z88P**:



(Windows: Plot program Z88P, undeflected structure. Looks similar on UNIX computers)

Now you've got some work: you must read off the node numbers for the boundary conditions in Z88P. We have to decide how to support the plate. We'll choose "cutting edges", i.e. the boundaries are supported by a "bezel" above and below. This allows angular movement crosswise to the bezels, but fixture in direction of the bezels.

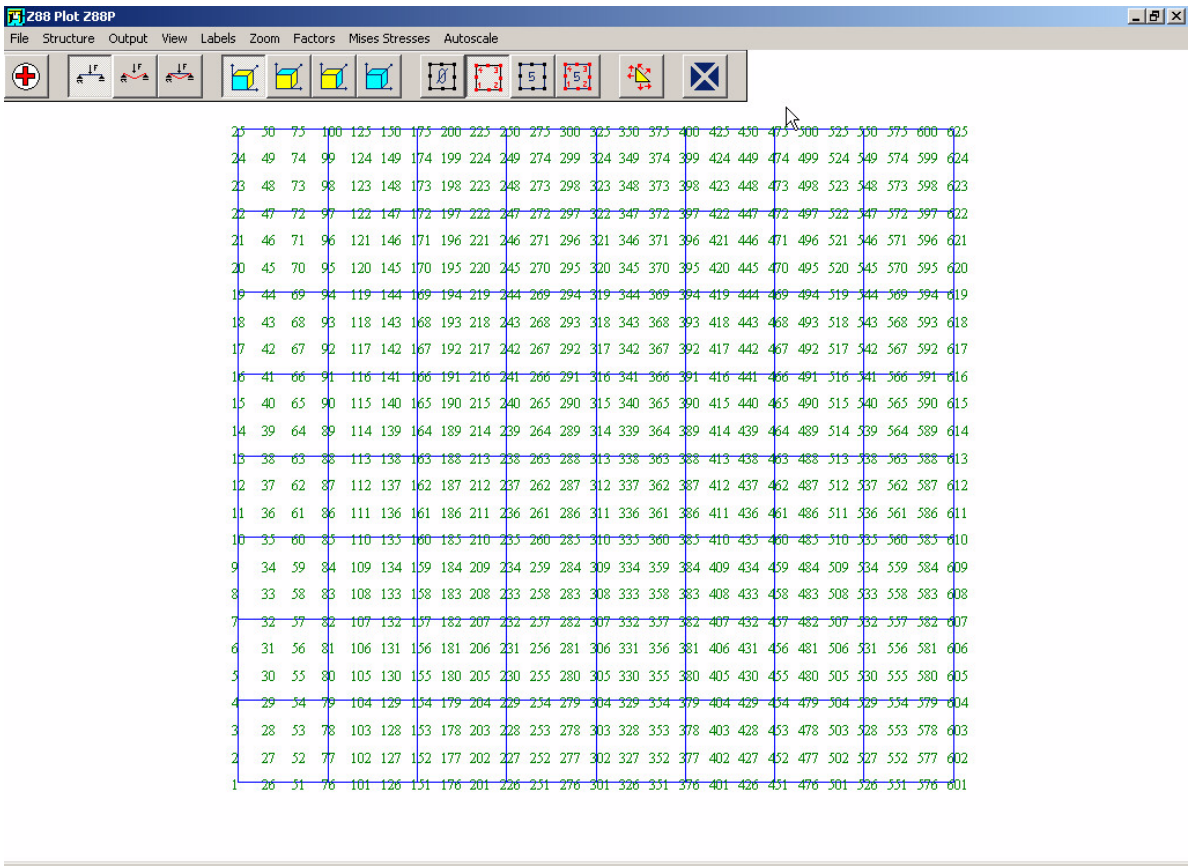


If you want to support the boundary in front, i.e. running in X direction, with cutting edges, then you must fix the degree of freedom 1 (the Z direction) and the degree of freedom 3 (the rotation around the Y axis).

We've got 625 nodes in total. Which to support ? Good question ! In order to save some work

(seldom a good idea) we'll try to fix only the corner nodes of the elements, which lay on the boundaries. This nodes are

- left boundary: 1, 4, 7, 10, 13, 16, 19, 22, 25
- lower boundary: 1, 76, 151, 226, 301, 376, 451, 526, 601
- upper boundary: 25, 100, 175, 250, 325, 400, 475, 550, 625
- right boundary: 601, 604, 607, 610, 613, 616, 619, 622, 625



(Windows: read off the the nodes with Z88P. Looks similar on UNIX machines)

See the beginning and the end of the boundary conditions file Z88I2.TXT (if you are too lazy to do the work of entering the boundary conditions: *B18_2ROU.TXT*) :

```
68
1     1   2   0.
1     2   2   0.
1     3   2   0.
4     1   2   0.
4     2   2   0.
....
622   1   2   0.
622   2   2   0.
625   1   2   0.
625   2   2   0.
625   3   2   0.
```

We may now launch one of the solvers. Because the structure is really tiny, the Cholesky solver is the right choice. The displacement file Z88O2.TXT gives us the information for

node 313, which lies exactly in the middle of the plate:

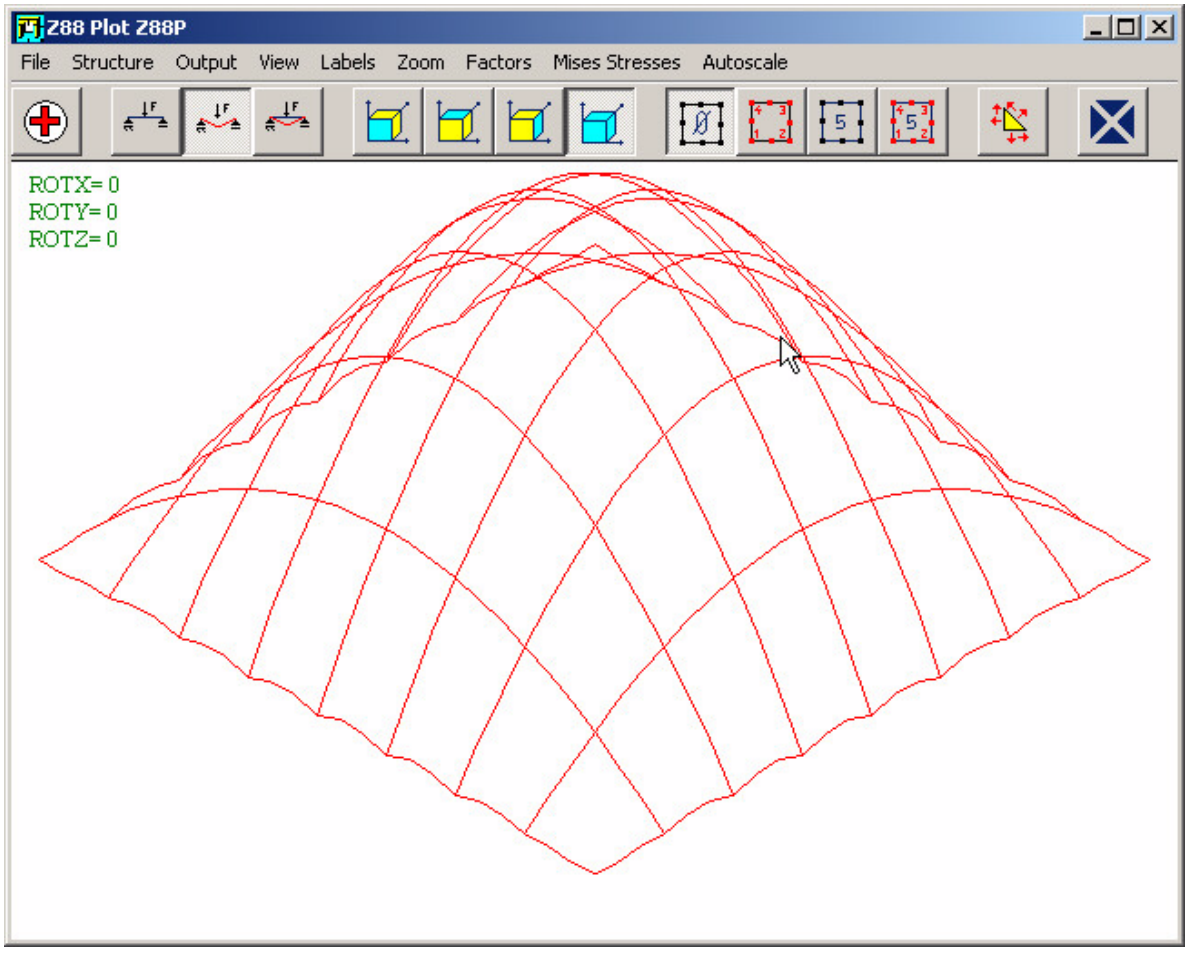
```
313    +1.1236511E+001    -2.1751298E-008    +2.1751298E-008
```

The deflection U2 (i.e. the rotation around the X axis) and U3 (i.e. the rotation around the Y axis) are zero, looks good. The deflection U1, i.e. w, is 11,24 mm. "Analytically" (this is also only an approximation for thin plates, ref. to the classical mechanics literature) one computes:

f= (0.71 * p * b**4) / (E * h**3)= (0,71 * 46,42 * 500**4) / (206.000 * 100**3) = 10 mm

This results in a variety of (10 - 11,24) / 10 * 100 = 12%.

Here's why. Firstly, the analytical formulae in the literature are thin plates of the Kirchhoff type neglecting the shear forces, secondly, this formulae were won with series expansion and thirdly, we could truely put some more work into a better formulation of the boundary conditions. Here's how our plot looks with a magnification factor of 50:



See how the boundaries raise between the corner nodes? Guess we must swallow the bitter pill and support all the nodes laying on boundaries (copy file *B18_2.TXT* to Z88I2.TXT). This results in:

w at node 313: 10,5 mm, variety to the analytical calculation about 5 % (the analytical calculation supplies thin plates and is not very exact here. This thin plates should feature a thickness of about 1/50, 1/100 or fewer of the main dimensions!)

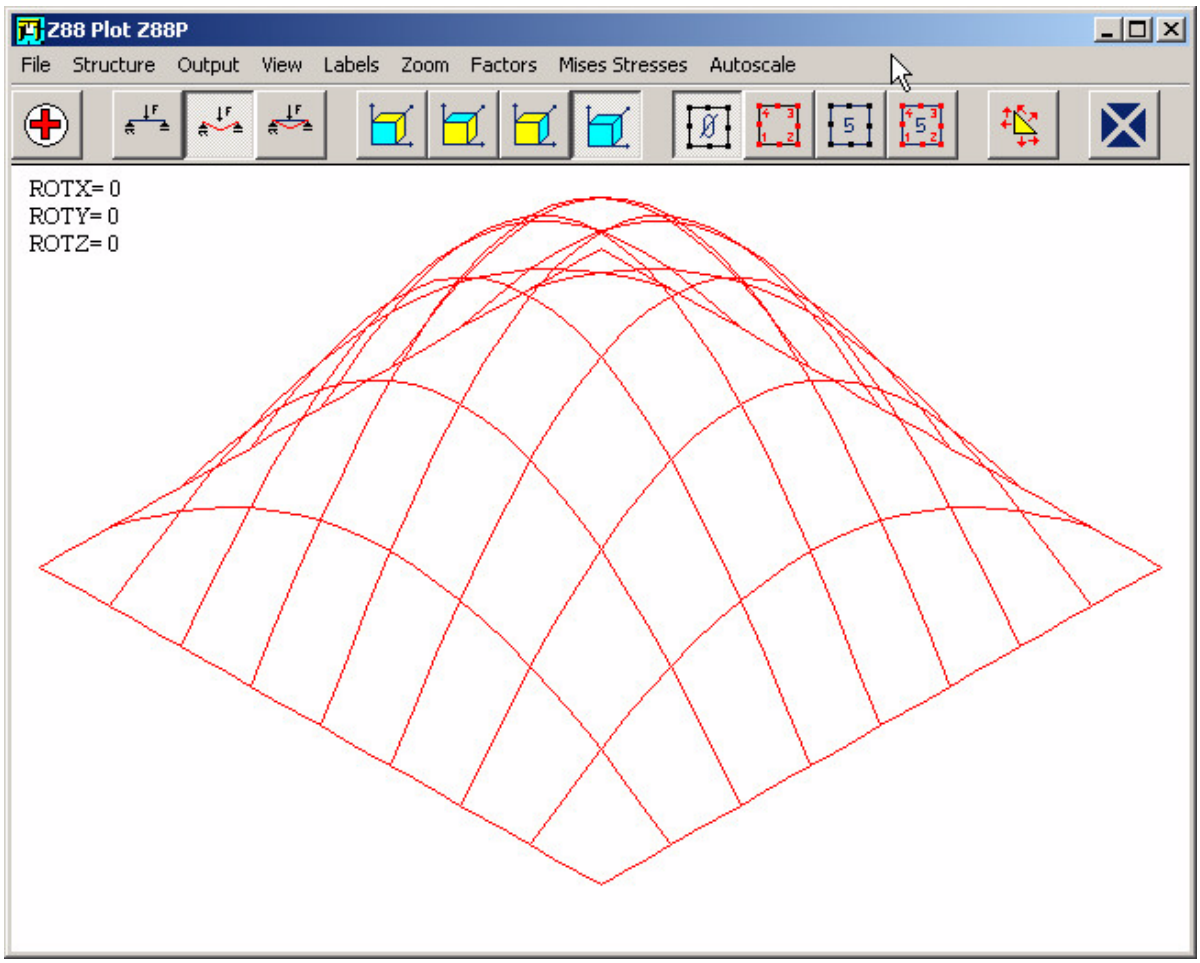
We may calculate the stresses "analytically":

$\sigma_x = \sigma_y$ = (1,15 * p * b**2) / h**2 = (1,15 * 46,42 * 500**2) / 100**2 = 1.335 N/mm**2

The stress parameter file Z88I3.TXT needs the following entries for computing the stresses in the corner nodes:

```
0     0     0
```

After running Z88D you may read off the stresses of node 313 from the elemente 28, 29, 36 or 37; it is the node with XX= 600 and YY= 600: $\sigma_x = \sigma_y$ = 1.334 N/mm**2.

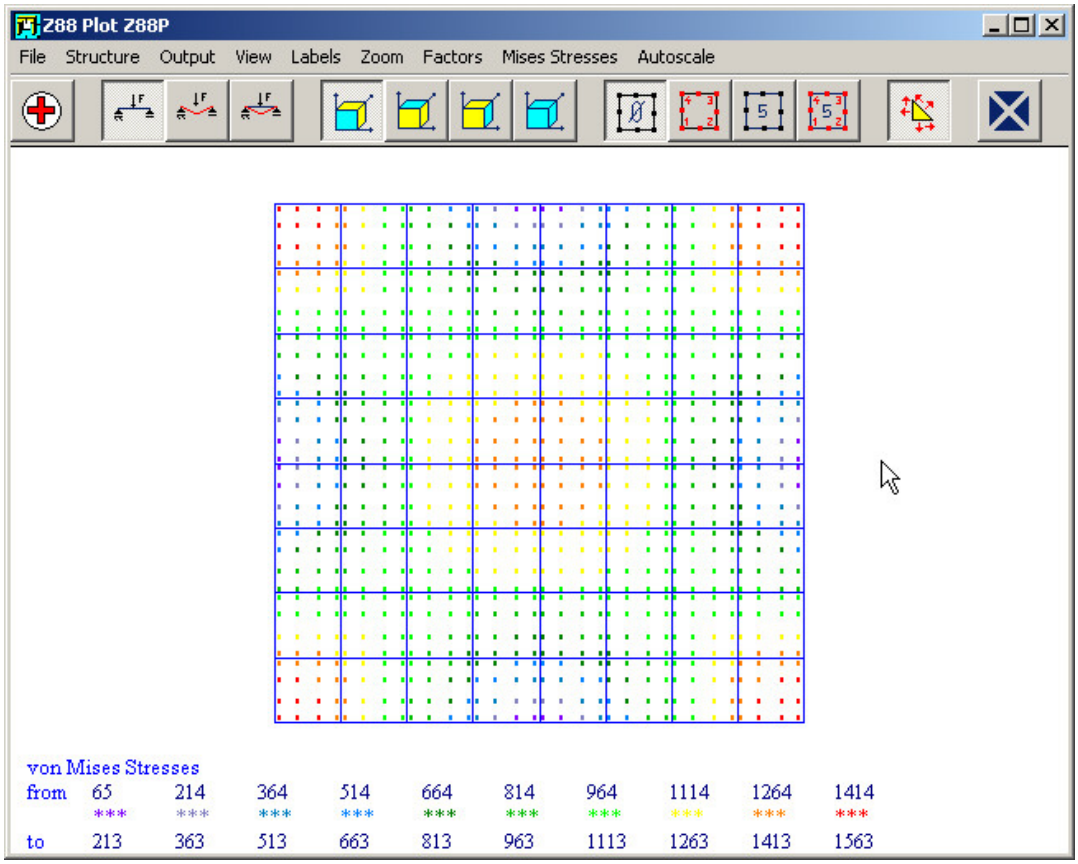


(Now the boundaries are supported properly)

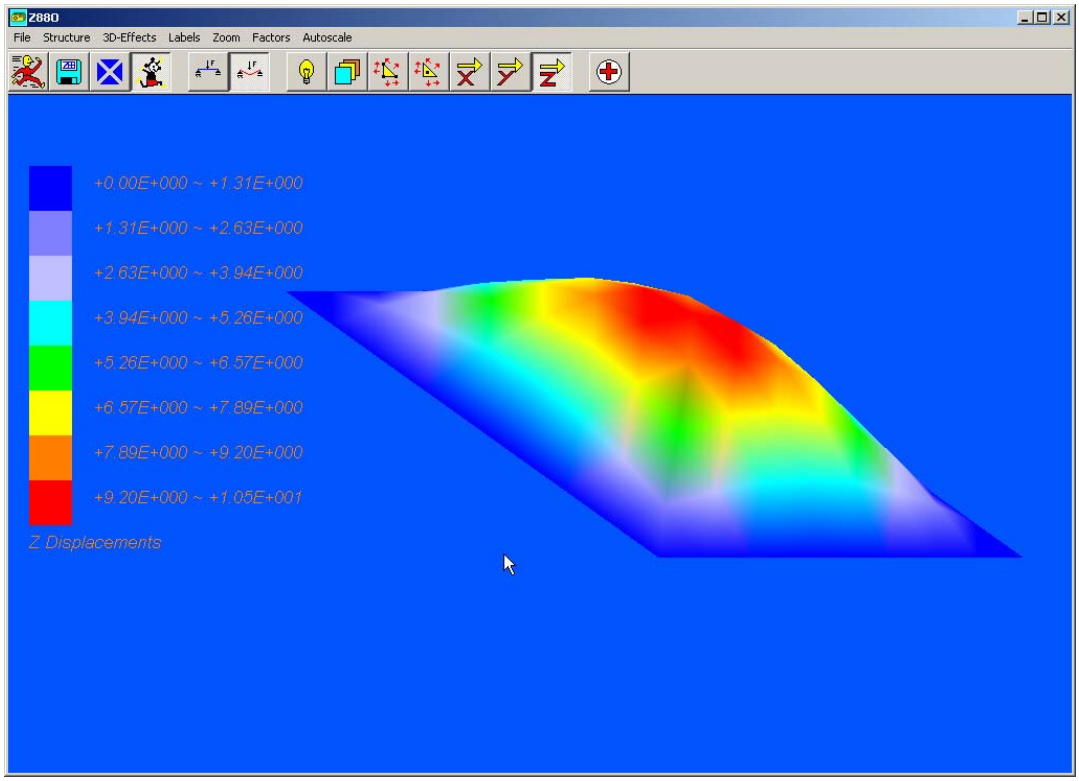
Finally, we'll compute the stresses in the Gauss points and, thus, adjust Z88I3.TXT as follows:

```
4     0     1
```

After a Z88D run we may look at the *von Mises* stresses:



(Windows: Plot of the von Mises *stresses in the 4 x 4 Gauss points. Z88P.)*



(Windows: Plot of the Z displacements. Z88O. Looks similar on UNIX machines)

Now you've got a small impression of plate calculation. Consult the devil (and Daniel Webster) when computing deflections and stresses for plates! I recommend parabolic tetrahedrons or hexahedrons in contrary for (thick) plate calculations, that means more input expense but the results are always save and free of suspicious interpretation constraints.