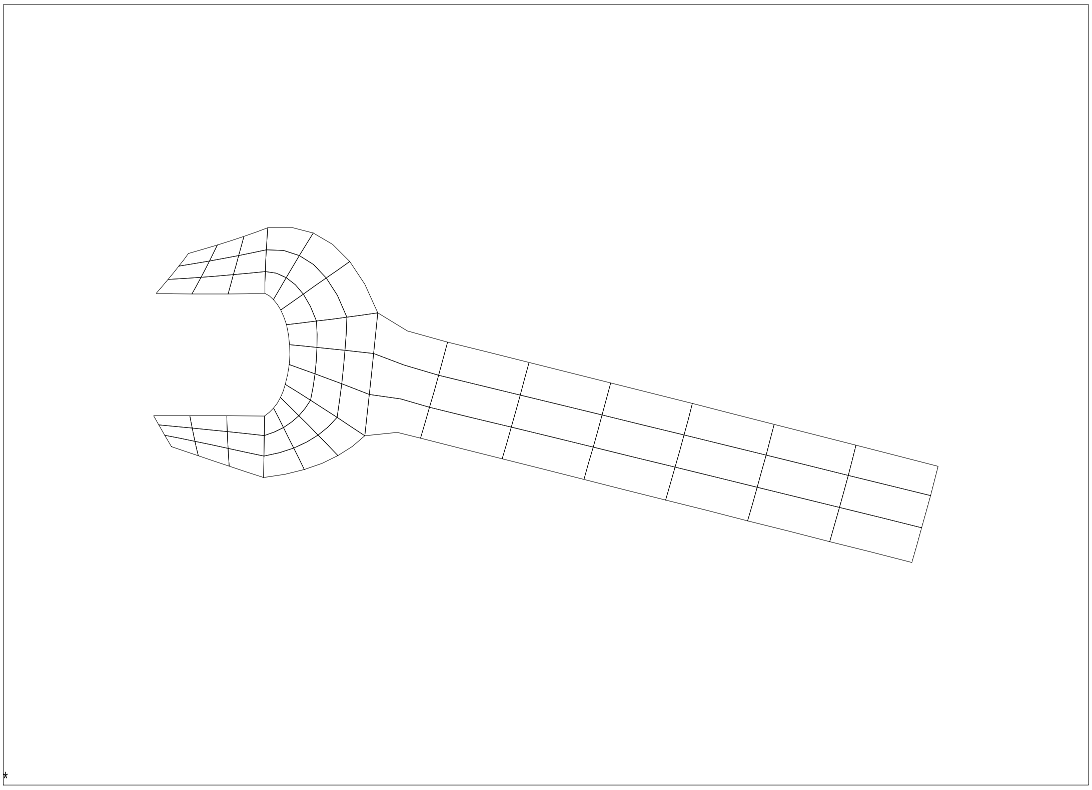
Z88

The compact Finite Element System



Version 13.0A

Z88

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

This Open Source version is 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 13.0A October 2009

WELCOMES TO Z88!

This Open Source version of Z88 for UNIX and Windows 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 PCs with Windows starting with Windows95 to Vista, LINUX, UNIX and Mac OS X 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 countless Windows, UNIX and LINUX installations worldwide. For static calculations in mechanical engineering and building & construction industries. Absolutely transparent for the user as input and output are handled by text files.

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.

The new version 13 looks quite similar to version 12 but the plot program is enhanced and two sparse matrix solvers with improved speed are included. In addition, now complete 64 bit support is supplied for Windows, LINUX, UNIX and Mac OS X. Thus, you may compute FE structures with millions of DOF with an ordinary PC.

Z88 for Windows comes ready-to-run for 32 bit XP and 32 bit Vista and for 64 bit Vista and Windows server, too. 32 bit and 64 bit RPMs are provided for LINUX. The Mac OS X solvers and utilities feature 64 bit support. You may compile Z88 for other UNIX/LINUX versions as Ubuntu and UNIX on your own computer.

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*
- *Rieg,F.; Hackenschmidt, R.: Finite Elemente Analyse für Ingenieure. Carl Hanser Verlag, München Wien 2009, 3rd edition (in German language)*

If you'll improve Z88 please give me your feedback. If you want to compile Z88 for UNIX any C compiler along with a *GTK+* and a *OpenGL* library should work fine. 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 and Intel.

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! Please see www.z88.de.

Bayreuth, September 2009

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

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:

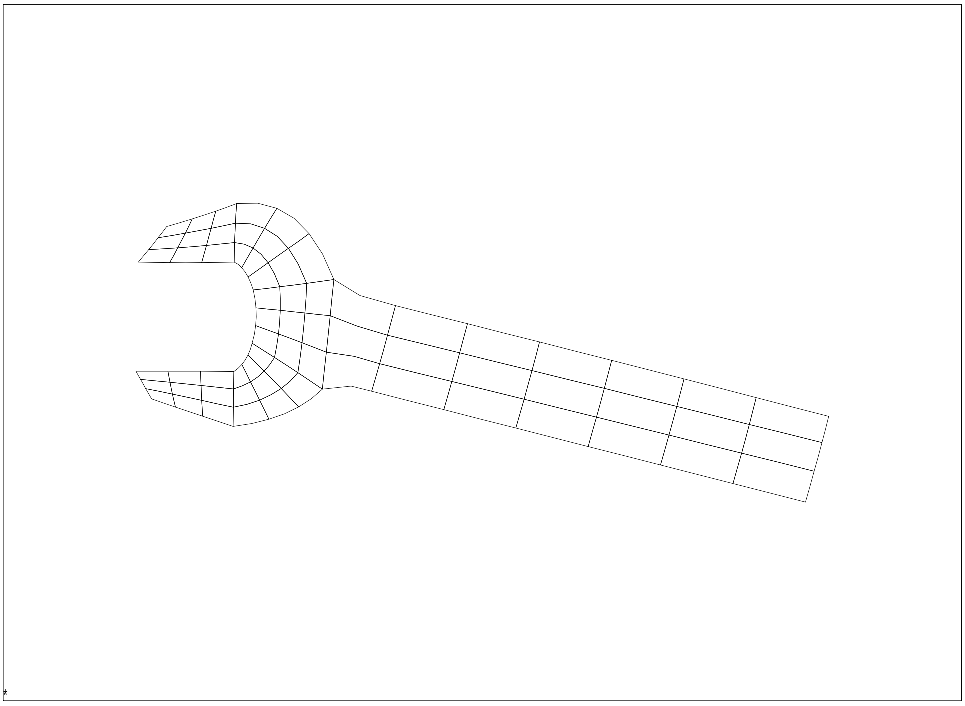
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.

<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 ELEMENTS PROGRAM Z88



1.1 GENERAL OVERVIEW OF 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* Windows, LINUX and Mac OS X programs, no emulation
+ Windows and UNIX programs use the same computing and graphic 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
+ Simplest installation: No subdirectories, no change of system files
+ Under UNIX: Automatic control and cumulative runs possible

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

In this manual, UNIX always means UNIX and LINUX and Mac OS X.

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!

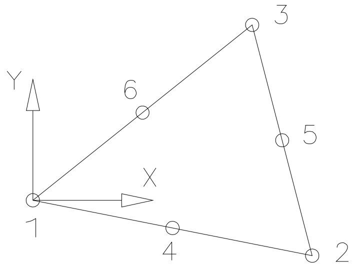
Summary of the Z88 element library:

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

Twodimensinal problems: Plane stress, plates, beams, trusses

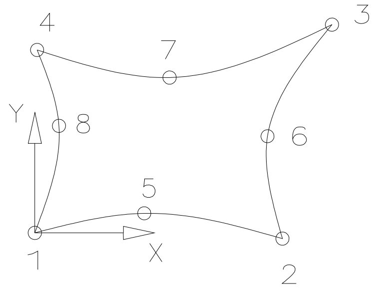
Plane Stress Triangle Element No. 3

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



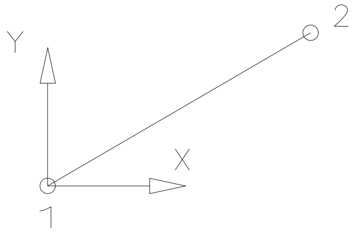
Plane Stress Isoparametric Element No. 7

- 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



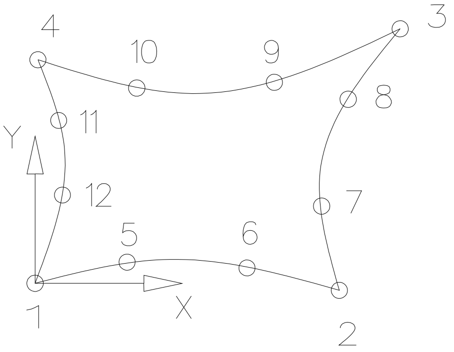
Truss No. 9

- 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



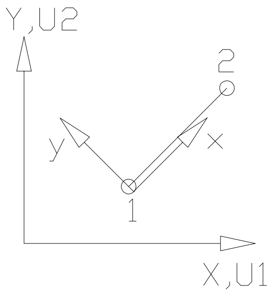
Plane Stress Isoparametric Element No. 11

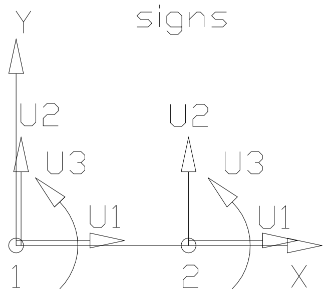
- Cubic Isoparametric Serendipity element
- Quality of displacements excellent
- Quality of stresses in the Gauss points excellent
- Quality of stresses in the corner nodes good
- Computing effort: Very high
- Size of element stiffness matrix: 24 × 24



Beam No. 13

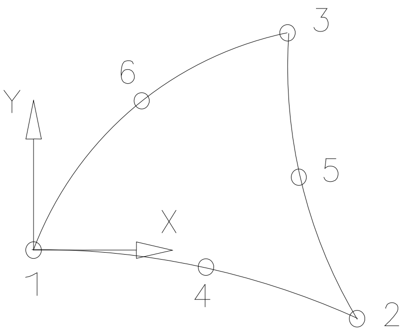
- Linear function for tensile stress, cubic function for bending stress
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 8 × 8





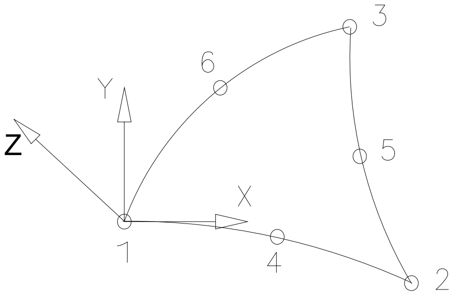
Plane Stress Isoparametric Element No. 14

- 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



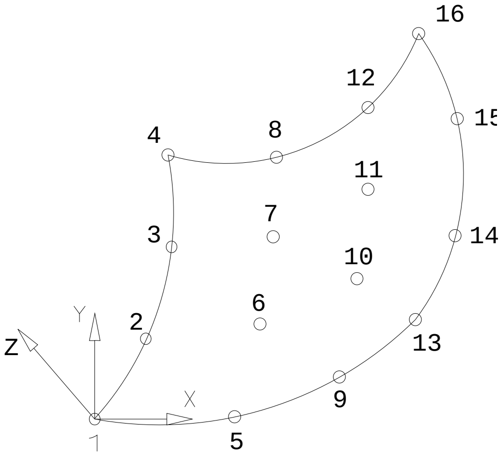
Isoparametric Plate Element No. 18

- 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



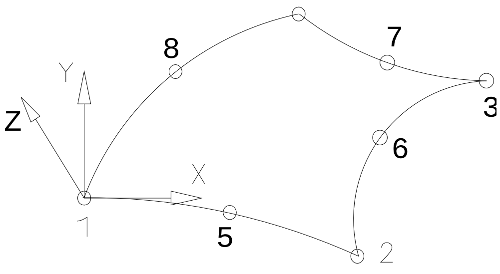
Isoparametric Plate Element No. 19

- 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

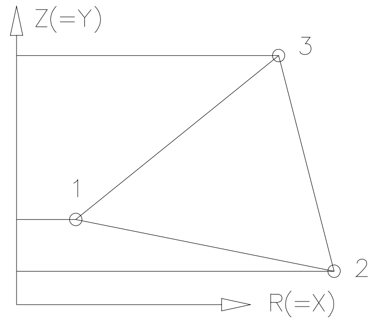
- Quadratic Isoparametric Serendipity element following Reissner-Mindlin's theory
- Quality of displacements very good
- Quality of stresses in the Gauss points good
- Quality of stresses in the corner nodes quite good
- Computing effort: medium
- Size of element stiffness matrix: 24 × 24



Axisymmetric problems:

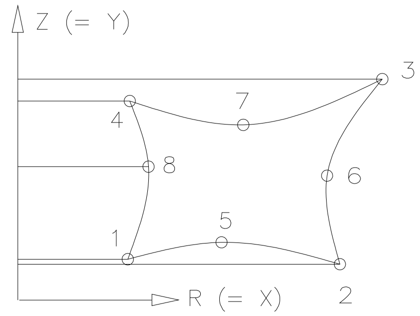
Torus No. 6

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



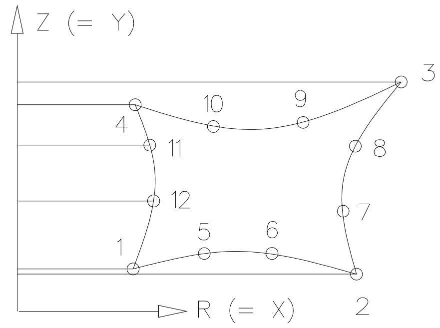
Torus No. 8

- 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



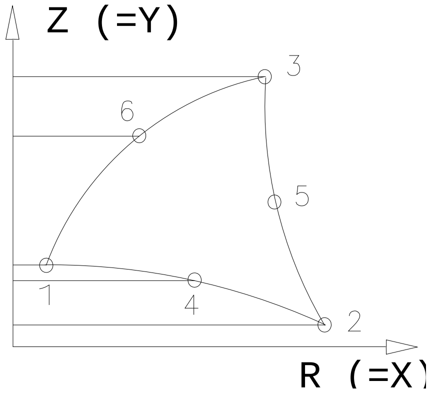
Torus No. 12

- 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



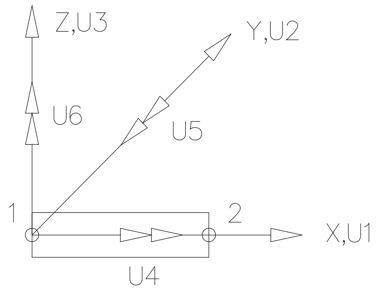
Torus No. 15

- 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

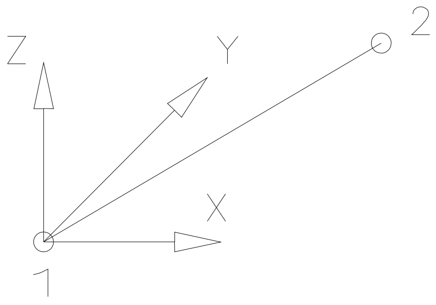
- Linear function for torsion and tensile stress, cubic function for bending stress
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 12 × 12



Spacial problems:

Truss No. 4

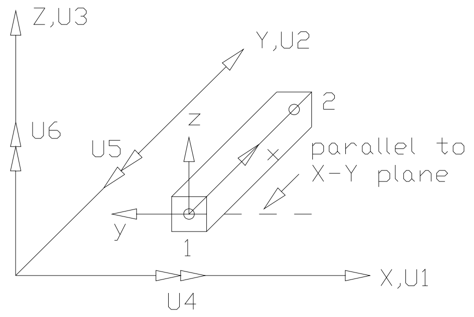
- Linear function
- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Minimal
- Size of element stiffness matrix: 6 × 6



Beam No. 2

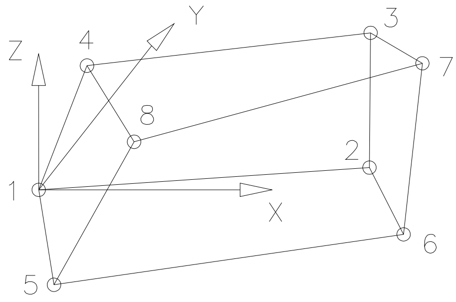
- Linear function for tensile stress, cubic function for bending stress

- Quality of displacements exact (Hooke 's law)
- Quality of stresses exact (Hooke ' s law)
- Computing effort: Low
- Size of element stiffness matrix: 12 × 12



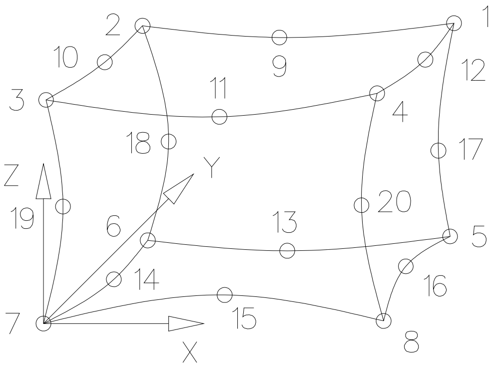
Hexahedron No. 1

- 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

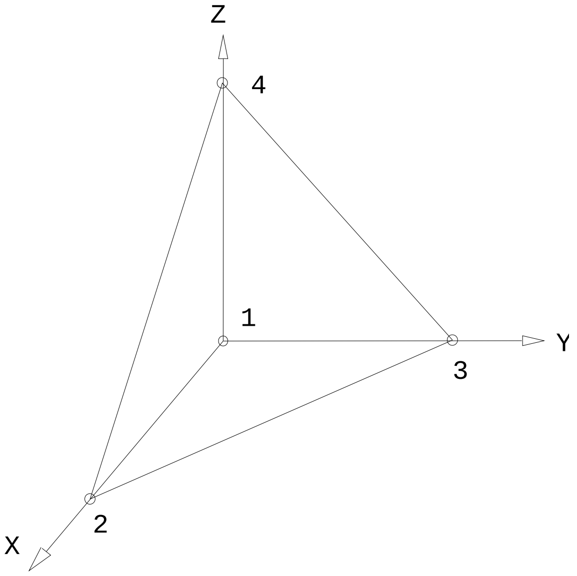
- Quadratic Isoparametric Serendipity element
- Quality of displacements very good
- Stresses in the Gauss points very good
- Stresses in corner nodes good
- Computing effort: extremely high
- Size of element stiffness matrix: 60 × 60



Tetrahedron No. 17

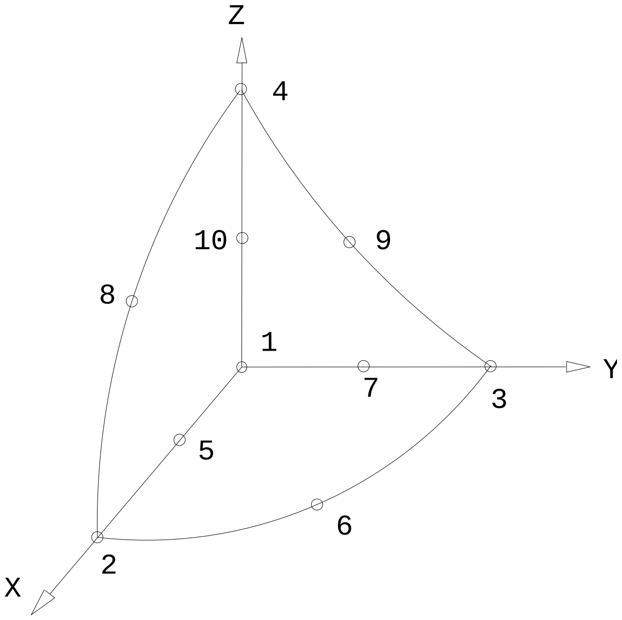
- Linear shape functions
- Quality of displacements bad

- 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. **Remark:** UNIX = LINUX, UNIX & Mac OS X.

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 and, if nessesary, the file for surface and pressure loads Z88I5.TXT. Basically, the Z88 input files can be created by CAD converter Z88X, by 3D-converter Z88G, by 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 three 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 ... 30,000 degrees of freedom.
- **Z88I1** and **Z88PAR**: This is a so-called direct sparse matrix solver with fill-in featuring two modules. Z88I1 computes the pointers for the storage scheme of the total stiffness matrix. Z88PAR uses the PARDISO solver and computes the stiffness matrices, addes the boundary conditions and solves the system of equations. This solver is very fast but uses very much dynamic memory. Z88I1/Z88PAR is your choice for medium-sized structures up to 150,000 DOF on ordinary 32 bit PCs. However, we’ve computed structures with ~ 1 million of DOF very fast using a computer featuring 32 (!) Gbyte of memory, 4 CPUs, 64 bit Windows version of Z88.
- **Z88I1** and **Z88I2**: This is a so-called sparse matrix 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. This solver needs some considerations but deals with structures with more than 100,000 DOF at nearly the same speed as the solvers of the large and expensive commercial FEA programs as our tests showed. In addition, a minimum of storage is needed. So, this solver is your choice for large structures with more than 150,000 … 200,000 DOF. Structures with ~ 5 million DOF are no problem for Z88I2 if you use a 64 bit operation system (Windows or LINUX or Mac OS X) along with the 64 bit version of Z88 and about 6 GByte of memory (with tricks i.e. compiling

Z88 with 8 Byte pointers and 4 Byte integers, 4 GByte will do). *This very stable and proved solver works always, thus, you may use it as your standard solver.*

II. The link to CAD programs

The **CAD converter Z88X** converts DXF files from CAD systems into Z88 input files (mesh generator input file Z88NI.TXT, general structure data Z88I1.TXT, boundary conditions Z88I2.TXT, surface and pressure loads Z88I5.TXT and stress parameters 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 **3D-converter Z88G** reads FEA input files following the COSMOS or the NASTRAN format and generates the Z88 input files Z88I1.TXT, Z88I2.TXT, Z88I5.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 **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 mesh generator for ordered meshes

The **mesh generator Z88N** reads the super structure data Z88NI.TXT and computes the general structure data Z88I1.TXT. In principle, the mesh 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/ Z88I2 or Z88I1/ Z88PAR 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/ Z88I2 or Z88I1/ Z88PAR must have run before. Z88E stores the nodal forces in Z88O4.TXT.

The **plot program Z88O** plots deflections and stresses. Operating in 3D mode, you may use wireframe or hidden line scenes or scenes with lighting. Z88O replaces the former plot programs Z88P and Z88O V12. The Windows version works with the *WinAPI* and *OpenGL,* the LINUX and Mac OS X version works with *GTK+* and *OpenGL*.

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 may 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.

LINUX/UNIX:
Launch the modules directly from a UNIX shell, from the Z88-Commander Z88COM, or, as an extended possibility, e.g. for large night runs, from a shell-script (*sh, bash, ksh* etc.). **You have all the unlimited liberties of the UNIX operating system.** All modules except Z88COM and Z88O can be started in text mode from consoles, but naturally also in an X window. As GKT+ programs the Z88-Commander Z88COM and the plot program Z88O are to start from an X-term. **LINUX**: For a convenient use of Z88, fire up your X-Window-manager, open an X-term and launch Z88COM. **Mac OS X**: Start the *Terminal* (Finder > Go > Utilities > Terminal) and launch Z88COM. Put Z88COM and the Terminal, 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, 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 3D- 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^2.

Note:

The Z88 input files read *always*:

+ Z88G.COS	COSMOS data file from 3D-CAD system for 3D-converter Z88G
+ Z88G.NAS	NASTRAN data file from 3D-CAD system for 3D-converter Z88G
+ Z88X.DXF	Exchange file for CAD programs and for CAD converter Z88X
+ Z88NI.TXT	Input file for the mesh generator Z88N
+ Z88I1.TXT	Input file (general structure data) for the solvers Z88F, Z88PAR, Z88I2
+ Z88I2.TXT	Input file (boundary conditions) for the solvers Z88F, Z88PAR, Z88I2
+ Z88I3.TXT	Input file (control values) for the stress processor Z88D
+ Z88I4.TXT	Input file (control values) for the sparse matrix solvers
+ Z88I5.TXT	Input file (surface and pressure loads) for the solvers Z88F, Z88PAR, 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 mesh generator file Z88NI.TXT, the general structure data file Z88I1.TXT, the boundary conditions file Z88I2.TXT, the file for surface and pressure loads Z88I5.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*	*mesh*	*creates*

				generator	
Pro/ENGINEER Pro/MECHANICA	Z88G.COS Z88G.NAS	**Z88G**	Z88I1.TXT, Z88I2.TXT, Z88I3.TXT, Z88I5.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, Z88I5.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. Make a screen dump by *Shift – Print* and paste it anywhere.
Use the printing routines of the **LINUX** operating system or print by using an editor like *gedit* or *OpenOffice*. Make screen dumps by *GIMP*.
On **Mac OS X** computers you may print the Z88 files by *TextEdit*, use the utility *screencapture* or make a screen dump by *Cmd – Shift – 3*.

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	mesh 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
Z88I5.TXT	ASCII	Input	surface and pressure loads	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 Z88O	No 1)	Yes	Yes
Z88O8.TXT	ASCII	Output	for internal use of Z88O	No 1)	Yes	Yes
Z88O.OGL	ASCII	Input	Color header file Z88O MS-Win	Possible	Yes	No
Z88.FCD	ASCII	Input	Fonts, Colors, Dimens. UNIX for Z88COM and Z88O	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
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 sparse matrix solvers

Remark: UNIX = LINUX, UNIX and Mac OS X

1.2 HOW TO INSTALL Z88 FOR WINDOWS

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 CD or Internet 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. The included file *libguide40.dll* is used by Z88PAR. 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. 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 *editor* from *Start* > *Programs* > *Accessories*.

Assume 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 help:

Integrate your favourite Internet Browser into Z88. This may be *Firefox* or *MS Internet Explorer*. Note: The help files are stored on your hard disk. Thus, you don't need any internet connection when running Z88.

(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*, Firefox is called *firefox.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: *c:\Program Files\Internet Explorer*. Let us assume your previous PATH variable looks as follows:

```
H:\VisualStudio6\Tools\WinNT;C:\Hugo;
```

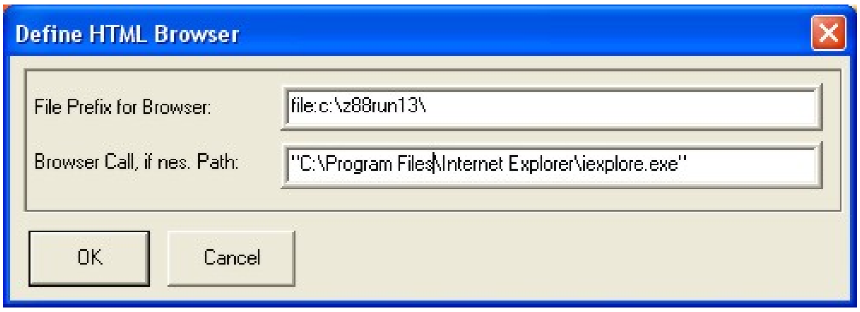
Separate the items by semicolons. And now:

```
H:\VisualStudio6\Tools\WinNT;C:\Hugo;c:\Program Files\Internet Explorer;
```

Logoff and login again.

2nd possibility: Enter path in Z88COM directly. If the path contains blanks then put the path into double quotes.

(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 Microsoft Internet Explorer and Firefox the prefix is *file:Z88 path*. Thus, for example:



File Prefix for Browser: file:c:\z88run13\

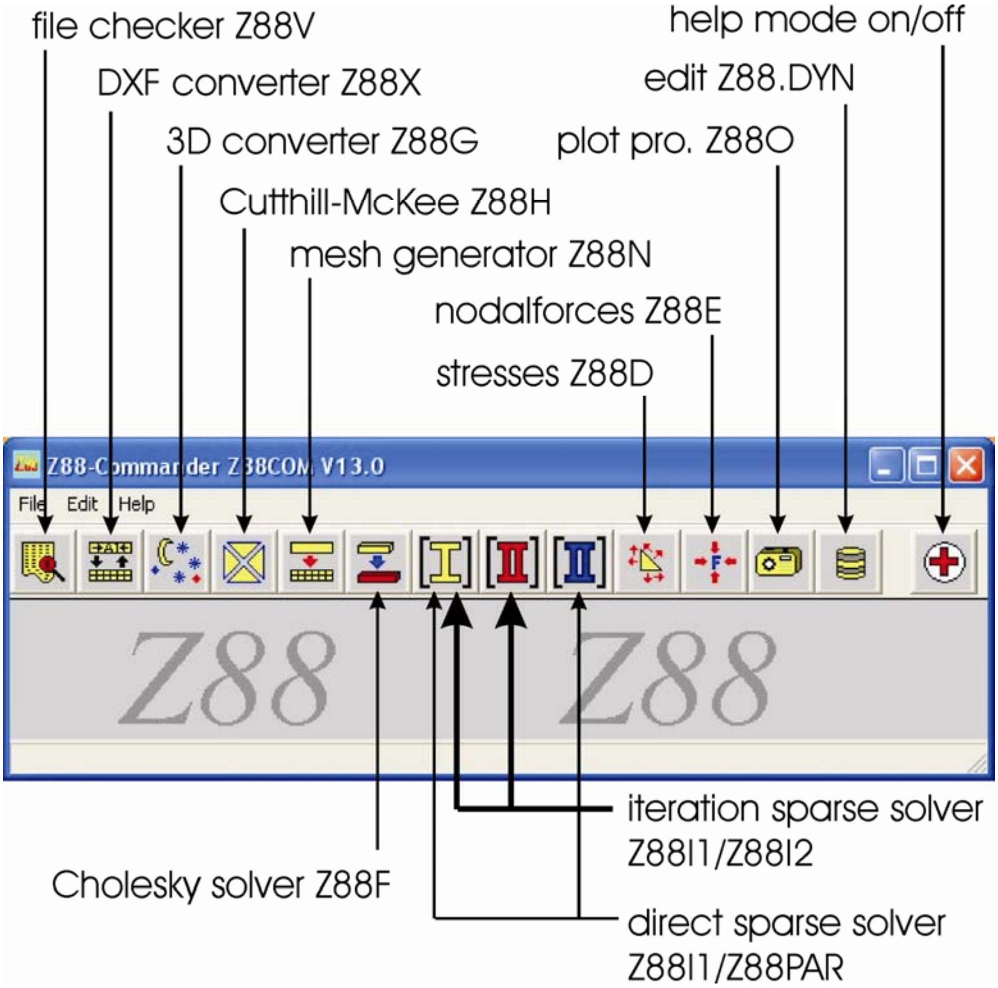
Browser Call, if nes. Path: "C:\Program Files\Internet Explorer\iexplore.exe"

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.



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:c:\z88run13
C:\Program Files\Internet Explorer\iexplore.exe

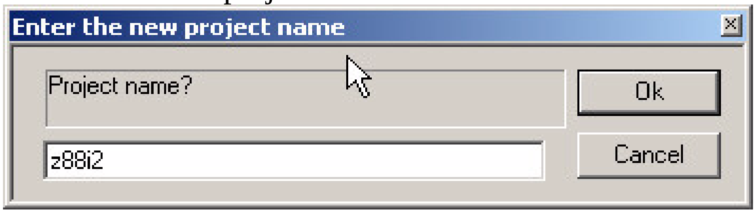
... And how do you remove Z88 ?

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 Studio 2005 and 2008) and Intel. 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 sparse matrix solver part 2 Z88I2 featuring the free LCC, but Microsoft Visual C++ compiler sessions are very similar :

1. Launch a new project.



Enter the new project name

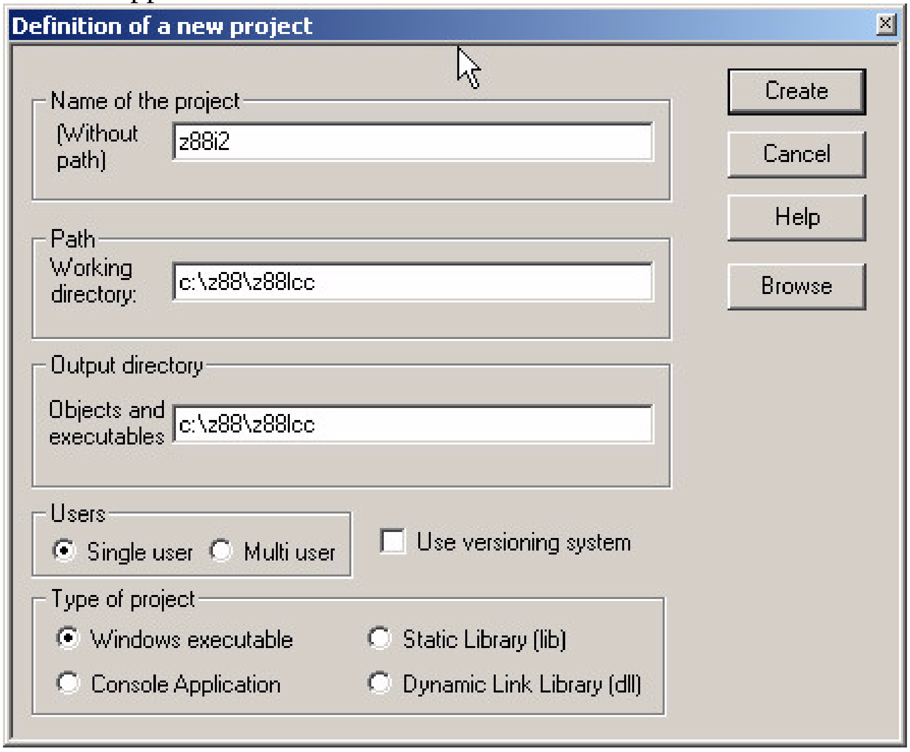
Project name? z88i2

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*.

Information

Do you want to use the wizard to generate the application skeleton?

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



Definition of a new project

Name of the project (Without path): z88i2

Path – Working directory: c:\z88\z88lcc

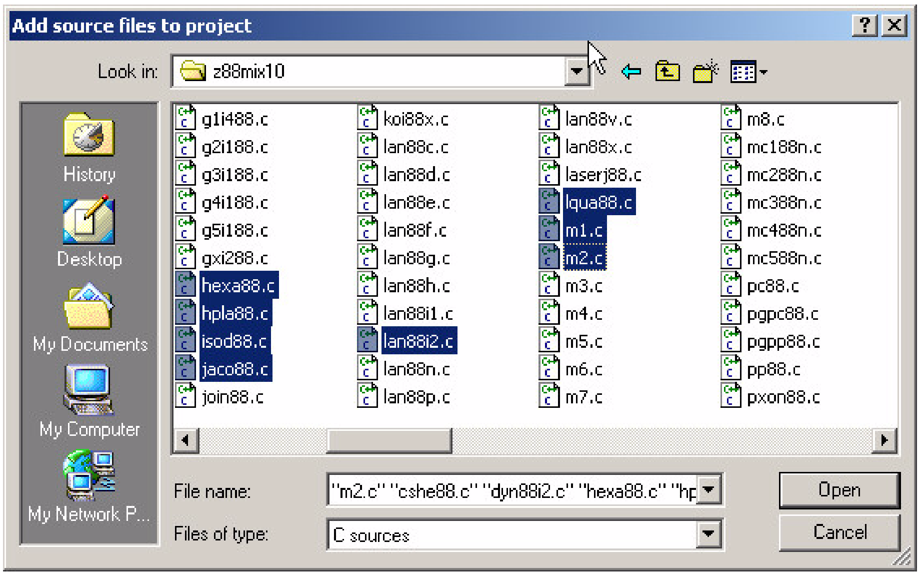
Output directory – Objects and executables: c:\z88\z88lcc

Users: ● Single user ○ Multi user

☐ Use versioning system

Type of project:
● Windows executable
○ Static Library (lib)
○ Console Application
○ Dynamic Link Library (dll)

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).



Look in: z88mix10

File name: "m2.c" "cshe88.c" "dyn88i2.c" "hexa88.c" "hp

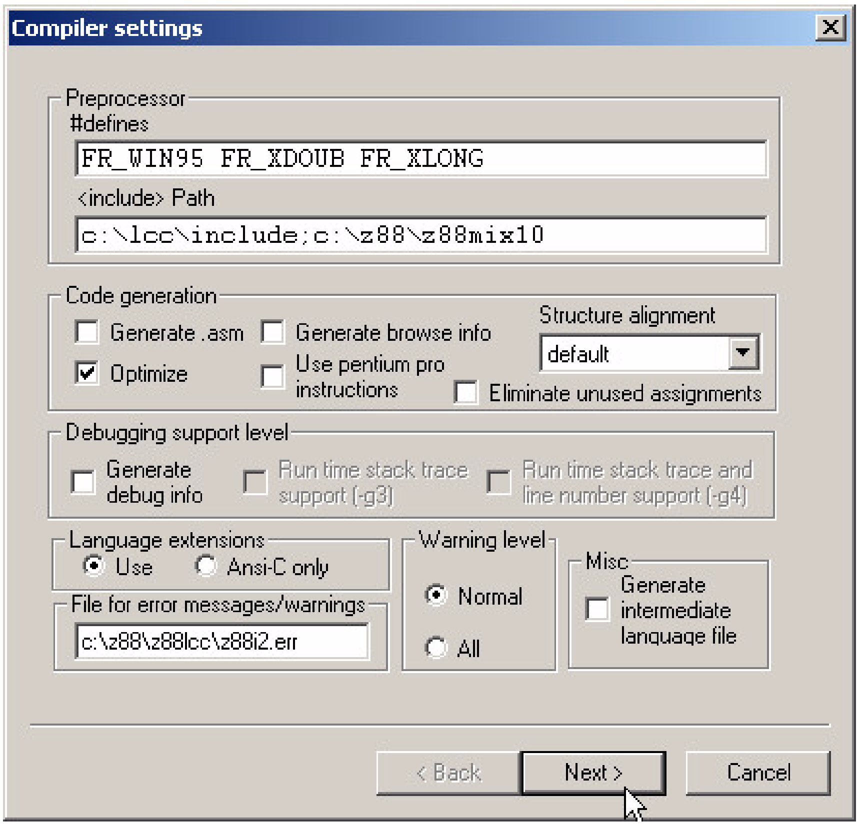
Files of type: C sources

5. <u>a most important step:</u>
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*) (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!



#defines: FR_WIN95 FR_XDOUB FR_XLONG

<include> Path: c:\lcc\include;c:\z88\z88mix10

☐ Generate .asm
☐ Generate browse info
☑ Optimize
☐ Use pentium pro instructions
Structure alignment: default
☐ Eliminate unused assignments

☐ Generate debug info
☐ Run time stack trace support (-g3)
☐ Run time stack trace and line number support (-g4)

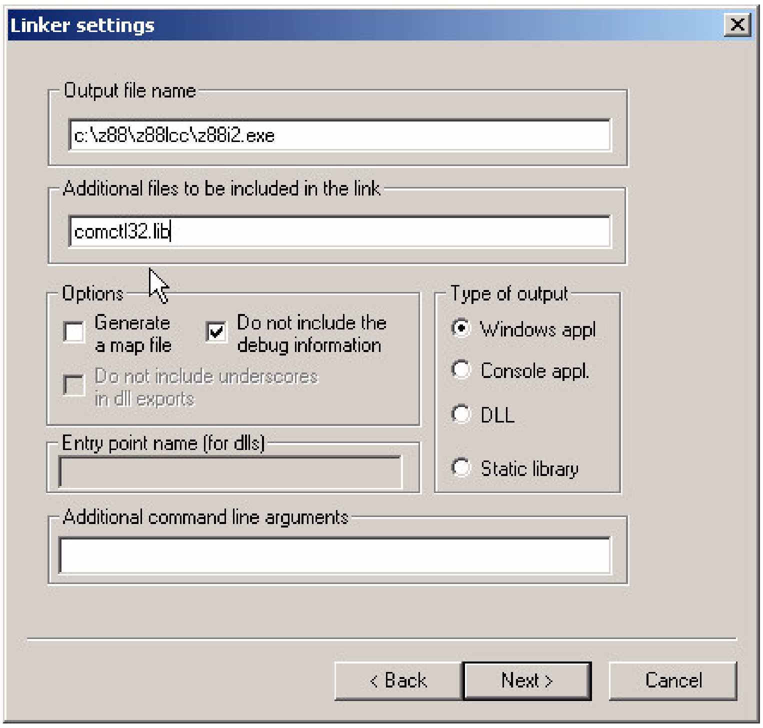
● Use ○ Ansi-C only

File for error messages/warnings: c:\z88\z88lcc\z88i2.err

● Normal
○ All

☐ Generate intermediate language file

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. For Z88O, add *opengl32.lib*.



Output file name: c:\z88\z88lcc\z88i2.exe

Additional files to be included in the link: comctl32.lib

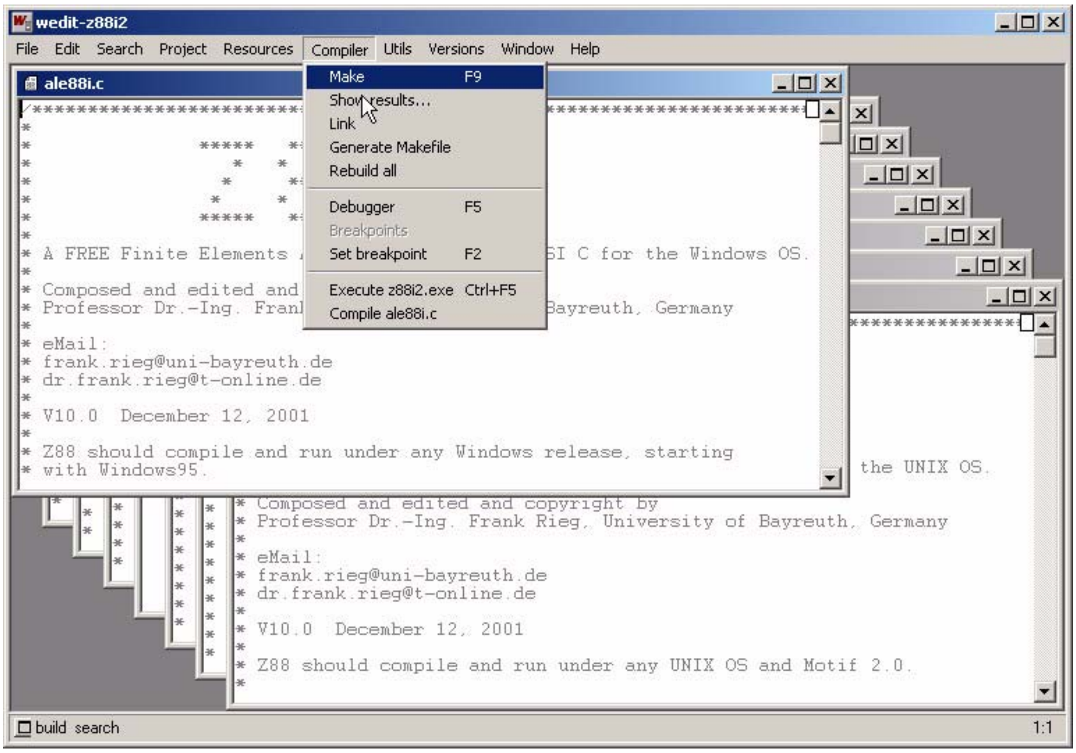
Options:
☐ Generate a map file
☑ Do not include the debug information
☐ Do not include underscores in dll exports

Entry point name (for dlls):

Type of output:
● Windows appl
○ Console appl.
○ DLL
○ Static library

Additional command line arguments:

7. *make project* or *rebuild all:*



```
* A FREE Finite Elements                              SI C for the Windows OS.
*
* Composed and edited and
* Professor Dr.-Ing. Fran                             Bayreuth, Germany
*
* eMail:
* frank.rieg@uni-bayreuth.de
* dr.frank.rieg@t-online.de
*
* V10.0  December 12, 2001
*
* Z88 should compile and run under any Windows release, starting
* with Windows95.
```

```
* Composed and edited and copyright by
* Professor Dr.-Ing. Frank Rieg, University of Bayreuth, Germany
*
* eMail:
* frank.rieg@uni-bayreuth.de
* dr.frank.rieg@t-online.de
*
* V10.0  December 12, 2001
*
* Z88 should compile and run under any UNIX OS and Motif 2.0.
*
```

8. Make sure to have the files *Z88.DYN*, *Z88COM.CFG* and *Z88O.OGL* 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.

9. Use the following compiler defines to define the number of bytes for every operating system and any precision:

	Number type	Win32	Win64	LINUX32	LINUX64 Mac OS X
Operating system	./.	FR_WIN95 FR_XWIN32	FR_WIN95 FR_XWIN64	FR_UNIX FR_LINUX	FR_UNIX FR_LINUX
	Pointer	4	8	4	8
FR_XINT	Integer	4	4	4	4
FR_XLONG	Integer	4	4	4	8
FR_XLOLO	Integer	4	8	8	8
FR_XDOUB	Float	8	8	8	8
FR_XQUAD	Float	8	8 / 16(1)	12 / 16(1)	12 / 16(1)

You may choose any precision by using the FR_ compiler defines. For example: You want to compile the sources for Vista 64 Bit. The integers should hold 8 Bytes memory each and the floats should hold 8 Bytes memory each, too:

FR_WIN95   FR_XWIN64   FR_XLOLO FR_XDOUB

Hint (1) : depending on your compiler you may give further compiler flags

These sources are needed:

Name	Sources, different for Windows and UNIX	common Windows and UNIX Sources	Libraries for Windows + UNIX
		The Kernel. Proper Defines: **FR_XINT, FR_XLONG, FR_XLOLO, FR_XDOUB, FR_XQUAD**	
z88f	z88f.c ale88f.c wrim88f.c easyfont.c tob88f.c z88f.rc **stop88f.c who88f.c clr88.c**	apla88.c bapla88.c bcshe88.c bhexa88.c bhpla88.c blqua88.c bqshe88.c bspla88.c bspur88.c btetr88.c choy88.c cshe88.c dyn88f.c hexa88.c hpla88.c isod88.c lan88f.c lqua88.c m1.c m2.c prfl88.c qshe88.c ri188.c spla88.c spur88.c tetr88.c wlog88f.c wria88f.c z88a.c z88b.c z88cc.c z88f.h	comctl32.lib **-lm -lc**
z88i1	z88i1.c ale88i.c wrim88i.c easyfont.c tob88i1.c z88i1.rc **stop88i.c who88i1.c clr88.c**	dyn88i1.c lan88i1.c ri188i.c w4y88i.c wlog88i1.c wria88i.c z88ai.c z88i.h	comctl32.lib **-lm -lc**
z88i2	z88i2.c ale88i.c wrim88i.c easyfont.c tob88i2.c z88i2.rc **stop88i.c who88i2.c clr88.c**	apla88.c bapla88.c bcshe88.c bhexa88.c bhpla88.c blqua88.c bqshe88.c bspla88.c bspur88.c btetr88.c cshe88.c dyn88i2.c hexa88.c hpla88.c isod88.c jaco88.c lan88i2.c lqua88.c m1.c m2.c prfl88.c qshe88.c r1y88i.c r4y88i.c spla88.c spur88.c tetr88.c wlog88i2.c z88bi.c z88ci.c z88i.h	comctl32.lib **-lm -lc**
z88par	no OpenSource	Because of copyright reasons no sources are included	libguide40.dll (included)
z88d	z88d.c ale88d.c wrim88d.c easyfont.c tob88d.c z88d.rc **stop88e.c who88e.c clr88.c**	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	comctl32.lib **-lm -lc**
z88e	z88e.c ale88e.c wrim88e.c easyfont.c tob88e.c z88e.rc **stop88e.c who88e.c clr88.c**	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	comctl32.lib **-lm -lc**
		The other Programs. Proper Defines: **FR_XINT, FR_XLONG, FR_XDOUB**	
z88n	z88n.c ale88n.c wrim88n.c easyfont.c tob88n.c z88n.rc **stop88n.c who88n.c clr88.c**	dyn88n.c join88.c lan88n.c mc188n.c mc288n.c mc388n.c mc488n.c mc588n.c rni88.c subn88.c wlog88n.c z88n.h	comctl32.lib **-lm -lc**
z88v	z88v.c ale88v.c g1i188.c g1i388.c g1i488.c g2i188.c g3i188.c g4i188.c g5i188.c gxi288.c gxi588.c tob88v.c z88v.rc **stop88v.c who88v.c clr88.c**	dyn88v.c lan88v.c wlog88v.c z88v.h	comctl32.lib **-lm -lc**
z88x	z88x.c ale88x.c wrim88x.c easyfont.c tob88x.c z88x.rc **stop88x.c who88x.c clr88.c**	dyn88x.c koi88x.c lan88x.c rea88x.c sub88x.c wlog88x.c wria88x.c z88fx.c z88tx.c z88x.h	comctl32.lib **-lm -lc**
z88g	z88g.c ale88g.c wrim88g.c easyfont.c tob88g.c z88g.rc **stop88g.c who88g.c clr88.c**	cosm88.c dnas88.c nast88.c lan88g.c wlog88g.c z88g.h	comctl32.lib **-lm -lc**
z88h	z88h.c ale88h.c wrim88h.c easyfont.c tob88h.c z88h.rc **stop88h.c who88h.c clr88.c**	lan88h.c rdy88h.c wlog88h.c z88h.h	comctl32.lib **-lm -lc**

Name	Sources, different for Windows and UNIX	common Windows and UNIX Sources	Libraries for Windows + UNIX
		Plot program and Z88 Commander. Proper Defines: FR_XINT, FR_XLONG, FR_XDOUB	
z88o	z88o.c ale88o.c wlog88o.c m11.c m13.c oglfont.c rogl88.c tob88o.c z88o.rc **cb88o.c m15.c rcoo88.c**	dyn88o.c lan88o.c m9.c m10.c m12.c m16.c oc88.c z88o.h	comctl32.lib opengl32. lib **`pkg-config - - cflags - -libs gtk+ gdkglext gdkglext-x11 gtkglext gtkglext-x11`**
z88com	z88com.c ale88c.c **cb88c.c rcol88.c** easyfont.c tob88c.c z88com.rc	lan88c.c wlog88c.c z88com.h	comctl32.lib **`pkg-config - - cflags - -libs gtk+ gdkglext gdkglext-x11 gtkglext gtkglext-x11`**

The Z88 executables and sources, Defines and Libraries(these files do exist in Windows only, these **files** do exist in UNIX only). **Remark**: UNIX = UNIX, LINUX and Mac OS X.

1.3 HOW TO INSTALL Z88 FOR LINUX, UNIX AND MAC OS X

1.3.1 LINUX installation for Fedora and SuSE

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:

Automatically for SuSE LINUX:
Insert the CD, change to directory */unix/rpm*, click the RPM file and let Yast install Z88. This works with Z88-RPMs downloaded from the Internet, too.
Manually for other distributions:
Check if these packages are installed: Firefox and Gedit. How to check? Do this:

- *rpm –q firefox*
- *rpm –q gedit*

Install these programs if necessary. They are part of RedHat (Fedora) and SuSE LINUX and most other distributions. You may exchange later the browser Firefox and the editor Gedit against programs of your choice. However, you can force the installation without these programs by use of `--nodeps`. Mount the CD, if nesserary:

- *mount –t iso9660 /dev/cdrom /cdrom*

If the mount point */cdrom* does not exist: go into the root directory and enter:

- *mkdir cdrom*

Go to */cdrom/unix/rpm* and install Z88 for 32 bit LINUX:

- *rpm –i z88-13.0-1.i586.rpm (default)*
- *rpm –i --nodeps z88-13.0-1.i586.rpm (forced)*

For 64 bit LINUX systems the installation command for 64 bit Z88 is as follows:

- *rpm –i z88-13.0-1.x86_64.rpm (default)*
- *rpm –i --nodeps z88-13.0-1.x86_64.rpm (forced)*

Launch Z88:
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. Because of legal reasons the sources for the solver Z88PAR are not GNU-GPL - but the very good solvers Z88F and Z88I1/Z88I2 are included, of course.

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 root, 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 *z88src.tar.gz*. Uncompress it:

- *gunzip z88src.tar.gz*
- *tar -xvf z88src.tar*

I suggest to set the file access rights to 777:

- *chmod 777 **

2nd Step: Compile Z88 for UNIX or LINUX:
You need: C compiler, make, X11, GTK+, OpenGL
Any UNIX-C or C++ compiler should work. I've tested the GNU *gcc* and the Intel C compiler.

- For LINUX: run *COMPILE.GCC32 or COMPILE.GCC64*
  These or similar libraries must be installed on your system:
  - xorg-x11-devel (X11 development library)
  - xorg-x11-Mesa-devel (OpenGL development library)
  - gtk+-devel (GTK+ development library)
  - gtkglext (GTK+ Widget for OpenGL)
- UNIX systems: modify one of the Makefiles (**.mk.**) and one of the *COMPILE.** files.
- The following Makefiles are included:

	for the solver programs	for the other programs	for the GTK+ programs
LINUX 32-Bit	*z88.mk.kernel.gcc32*	*z88.mk.other.gcc32*	*z88.mk.gtk.gcc32*
LINUX 64-Bit	*z88.mk.kernel.gcc64*	*z88.mk.other.gcc64*	*z88.mk.gtk.gcc64*

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

This is possible for the solver modules Z88F, Z88I1 and Z88I2 along with their subroutines and the stress processor Z88D and the nodal force program Z88E. For the rest of the Z88 modules (Z88COM, Z88G, Z88H, Z88N, Z88O, Z88V) only FR_XDOUB is implemented because it makes no sense to run the plot program Z88O or the DXF converter Z88X with extended precision. Anyway, 64 Bit integers and pointers are possible. Therefore, *three* makefiles do exist and it's a good idea to run them one after another.
For example: *make –f z88.mk.kernel.gcc32*

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. *Firefox:* 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:

- *Firefox*: *file:///home/frank/z88run13/*, assuming that the Z88 HTML, the GIF and the JPG files are located in the directory */home/frank/z88run13*

You may easily find out the prefix for your browser if you start it from an X-term with a Z88 HTML file, e.g. *firefox file:///home/frank/z88run13/e88ix.htm*

The help system is easy to use: Clicking the *Help* button invokes context sensive online-help: 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*. *gedit* is quite nice, too. Edit *Z88.FCD.*

5th step: Adjust system variables:
Allow the Z88 programs to start in your favourite directory. Modify your file *.profile, .bashrc, .bash_profile* :

- *PATH=$PATH:.* (mind the dot after the colon!)
- *export PATH*

If you’ve got a „GERMAN“ or another non-angloamerican LINUX then adjust the system variable LANG in every case in *.profile, .bashrc, .bash_profile* to:

- *LANG=C*
- *export LANG*

Otherwise, the dots in the Z88 files are wrong interpreted and the plot program Z88O prints fancy wrong colours. Why? Because the thousands in Great Britain and the USA are

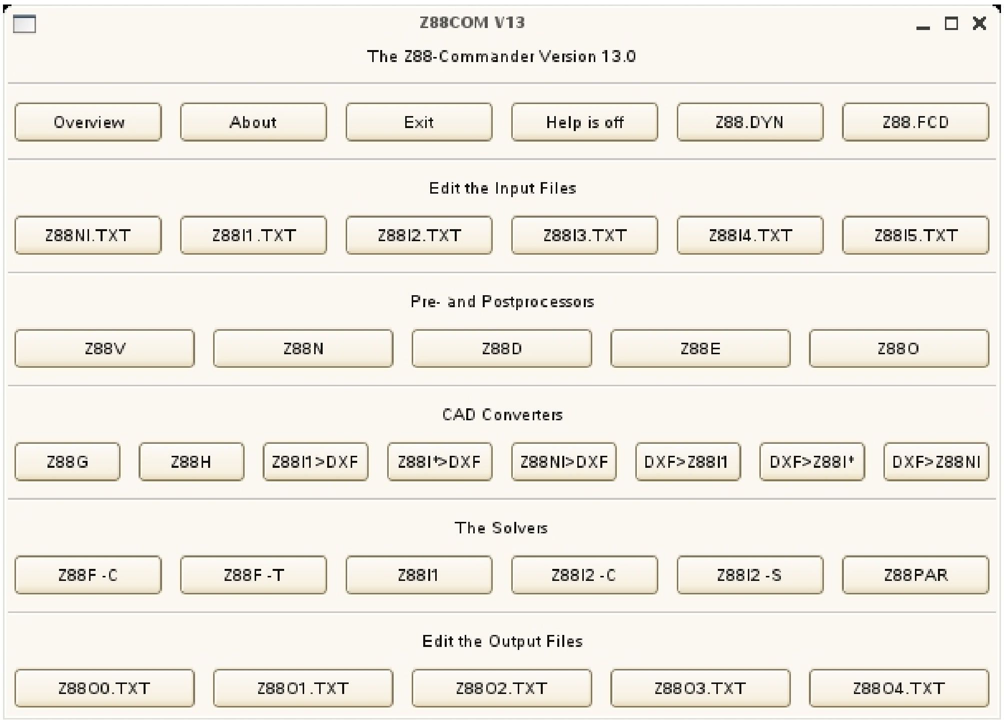
delimited by commas and the decimal point is the dot but in Germany and other European countries the thousands are delimited by dots and the decimal point is the comma!
LANG=C or LANG=en: 1.000 reads 1.   (correct!)
LANG=de   : 1.000 reads 1000.   (wrong!)
To make the variable LANG work, log off and log on again.

6th step: Adjust Z88's language:

Enter into the file Z88.DYN either the keyword ENGLISH or GERMAN. ENGLISH is default, thus, you may skip this step.

7th step: 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 program Z88O must be started on an X-Window surface like *gnome* or *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 launch 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 colours 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 Z88O cannot run without a correct *Z88.FCD*.

... And how do you remove Z88?

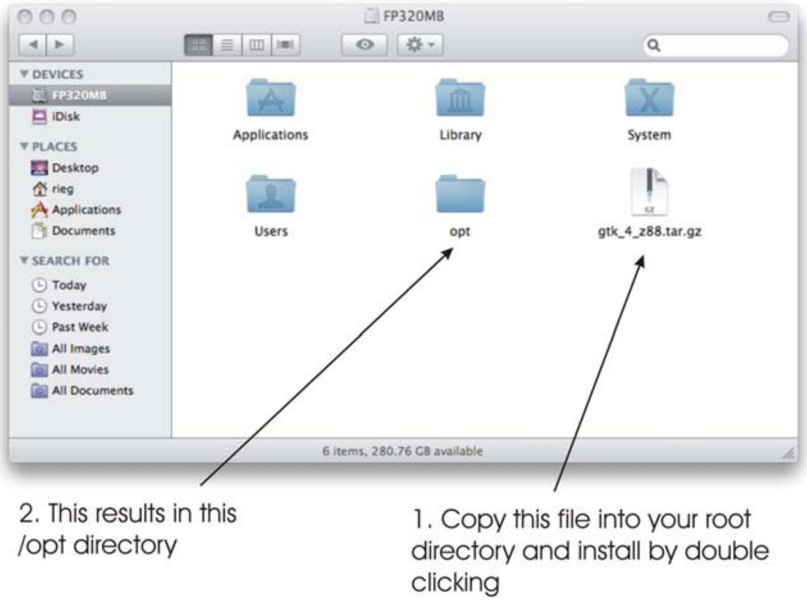
Simply delete all files in the directory containing Z88. Then delete the directory if necessary.

1.3.3 Installation for Mac OS X

Install Z88 V13.0A ready-to-run for Mac OS X

1st step: Install the GTK runtime libraries:
Z88 needs the GTK widget set for its window operations. While most LINUX systems feature GTK+ by default, Mac OS X does not have this toolkit. I compiled for you a GTK+ runtime system which contains all the features Z88 needs.

You need to have admin rights. Copy the file *gtk_4_z88.tar.gz* into the root directory /. Install by double clicking. This will create a directory */opt* . If *opt* is created in the folder *Download* then copy *opt* into the root directory.



2. This results in this /opt directory

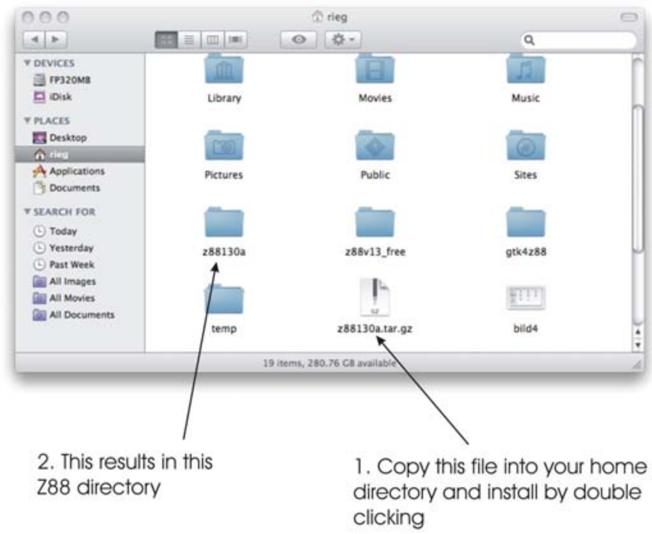
1. Copy this file into your root directory and install by double clicking

Installing the GTK+ Libraries for Z88

Or you may use the *tar* command in the *Terminal* (Finder > Go > Utilities > Terminal)*:*
- cd /
- *sudo tar –xvzf gtk_4_z88.tar.gz*

If there a folder /opt exists before installing the Z88 GTK Runtime: Rename this still existing folder /opt to, let's say, /opt_org. Rename the folder /opt created by gtk_4_z88.tar.gz to, for example, /opt_z88. Create a soft link depending from the folder you want to use:
your genuine folder /opt: *sudo ln –s opt_org opt*
the folder /opt for Z88: *sudo ln –s opt_z88 opt*
You may remove later this link by *sudo rm opt* and create a new one.

2nd step: Install Z88:
Copy the file *z88130a.tar.gz* into your home directory and install by double clicking. This will create a new directory z*88130a* .



2. This results in this Z88 directory

1. Copy this file into your home directory and install by double clicking

Installing Z88

Or you may use the *tar* command in the *Terminal* (Finder > Go > Utilities > Terminal)*:*

- *cd*
- *tar –xzvf z88130a.tar.gz*

3rd step: Adjust system variables:

Allow the Z88 programs to start in your favourite directory. Modify your file *.bash_profile* in your home directory by using the editor *nano. Or, if .bash_profile* does not exist, create one by using the small editor nano in the Terminal: *nano .bash_profile.* Enter:

- *export PATH=$PATH:.* (mind the dot after the colon!)

Adjust the system variable LANG in every case in *.bash_profile* to:

- *export LANG=C*

To make the variables work, stop Terminal and start Terminal again.

4th step: Adjust Z88's language:

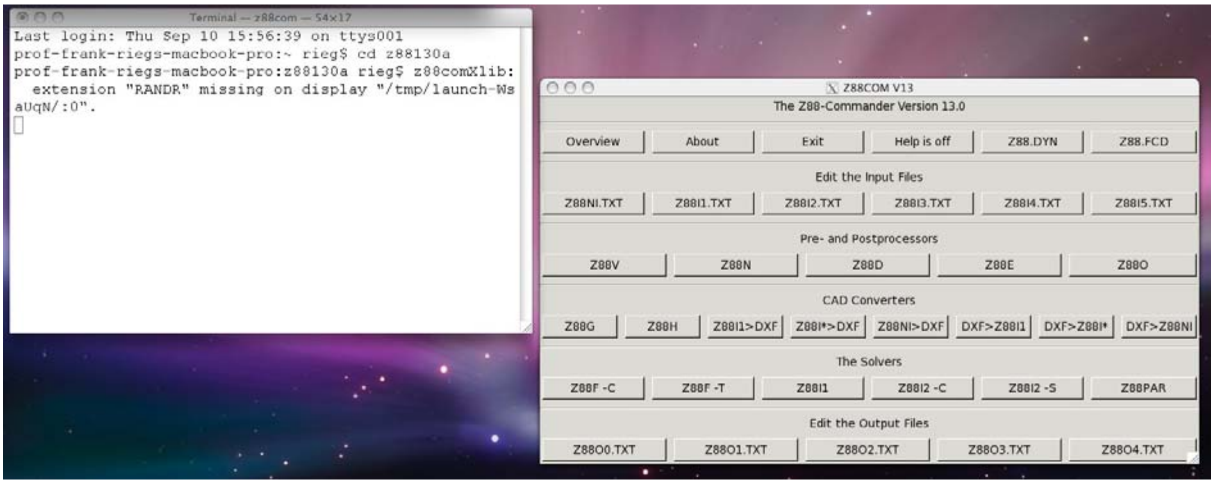
Enter into the file Z88.DYN either the keyword ENGLISH or GERMAN. ENGLISH is default, thus, you may skip this step.

5th step: Run Z88:

Start the Z88 Commander Z88COM from the *Terminal* (Finder > Go > Utilities > Terminal) in your Z88 directory z88130a by

- *z88com*

Starting Z88COM for the first time may need some time because the GTK+ system is loaded into memory. Put Z88COM and the Terminal, which started Z88COM, side-by-side or over-and-under to see both. The Terminal is used for console input/output for the text-mode programs Z88F, Z88I1, Z88I2, Z88N, Z88D, Z88E, Z88X, Z88G, Z88H, Z88V. Don't worry about the warning "Xlib: missing extension RANDR.."



```
Last login: Thu Sep 10 15:56:39 on ttys001
prof-frank-riegs-macbook-pro:~ rieg$ cd z88130a
prof-frank-riegs-macbook-pro:z88130a rieg$ z88comXlib:
  extension "RANDR" missing on display "/tmp/launch-Ws
aUqN/:0".
```

The Apple editor *TextEdit* (Finder > Go > Application > TextEdit) is predefined in Z88.FCD – be shure to operate *TextEdit* in plain text mode (TextEdit > Preferences > Plain Text)! The Apple browser *Safari* for the online help is predefined in Z88.FCD. You may change later both the editor and browser in the file *Z88.FCD* to, for example, *emacs* and *Firefox* (if installed). Be shure to put commands with more than one token as *open –a safari* in *:

open –a safari

If you are not pleased with my choice of colours 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 Z88O cannot run without a correct *Z88.FCD*.

... And how do you remove Z88?

Simply delete all files in the directory containing Z88. Then delete the directory if necessary.

Compiling Z88 V13.0A for Mac OS X

Only if you want to add improvements to Z88 you will need to compile the package. Because you need to install a couple of development tools you should only proceed if you have some good UNIX knowledge.

1st step: Install development tools und libraries

proceed in this order:

1) *Xcode (Mac Installation DVD or Apple Internet page)*
2) *MacPorts (Internet)*
3) *gtk2 (Internet)*
4) *gtkglext (Internet)*

2nd step: Compiling Z88

This works similar to chp.1.3.2:

- *make –f z88.mk.kernel.gccmac*
- *make –f z88.mk.other.gccmac*
- *make –f z88.mk.gtk.gccmac*

Use the command COMPILE.MAC.

Proceed with the 3rd step from page 38.

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 mesh generator (NET START, NET END), a common section for all modules (COMMON START, COMMON 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 and Z88I2 with the Filechecker Z88V.

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:

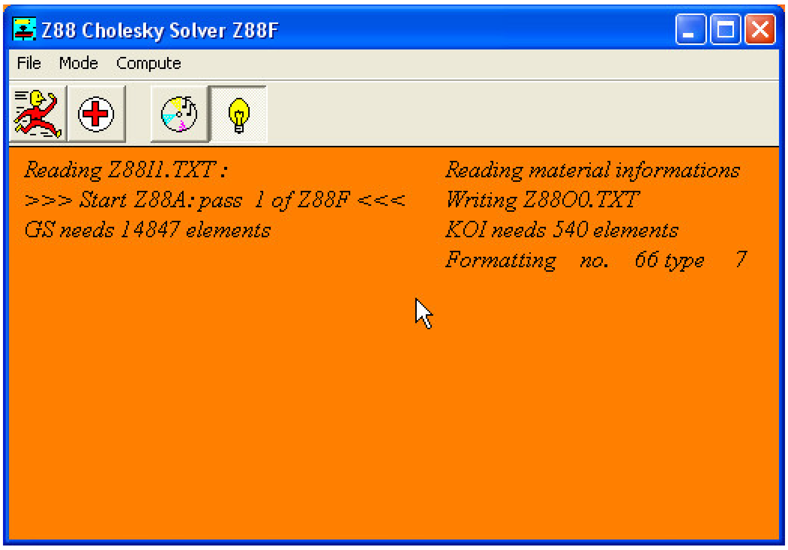
Windows: *Z88F > Mode > Test Mode , Compute > Go*
UNIX: *z88f -t (console) or Z88F -T (Z88COM)*

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: *Z88F > Mode > Test Mode , Compute > Go*
UNIX: *z88f -t (console) or Z88F -T (Z88COM)*

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

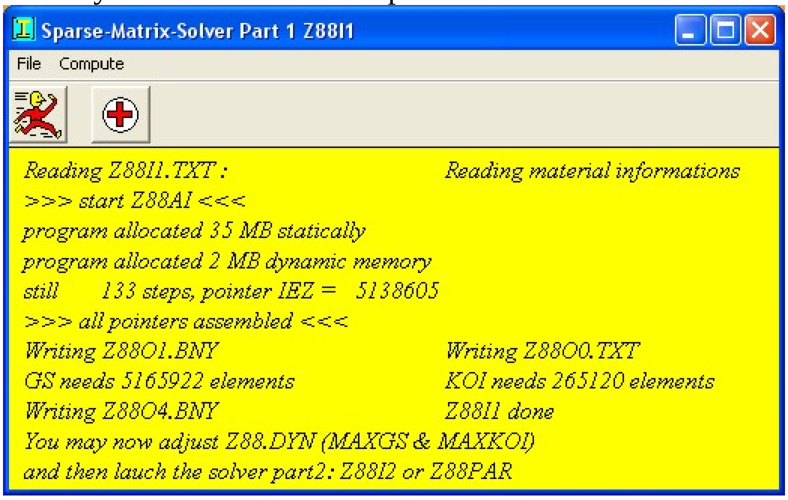


```
Reading Z88I1.TXT :                     Reading material informations
>>> Start Z88A: pass  1 of Z88F <<<     Writing Z88O0.TXT
GS needs 14847 elements                 KOI needs 540 elements
                                        Formatting   no.   66 type    7
```

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

The Sparse Matrix Solvers Z88I1/Z88I2 and Z88I1/Z88PAR:

There's no test mode available because the first part of the iteration solver Z88I1 detects the memory needs for the second part Z88I2 or Z88PAR:



```
Reading Z88I1.TXT :                     Reading material informations
>>> start Z88AI <<<
program allocated 35 MB statically
program allocated 2 MB dynamic memory
still      133 steps, pointer IEZ =    5138605
>>> all pointers assembled <<<
Writing Z88O1.BNY                       Writing Z88O0.TXT
GS needs 5165922 elements               KOI needs 265120 elements
Writing Z88O4.BNY                       Z88I1 done
You may now adjust Z88.DYN (MAXGS & MAXKOI)
and then lauch the solver part2: Z88I2 or Z88PAR
```

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

However, the procedure for the sparse matrix solvers is quite tricky because you must define memory MAXIEZ for the assembly of the sparse matrix. There is no way to pre-determine the needed memory but Z88I1 tells you if MAXIEZ was too small. Then, increase MAXIEZ in Z88.DYN and run Z88I1 again. 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 and Z88PAR.

3rd: if Z88I1 stopped because of lack of MAXIEZ increase MAXIEZ in Z88.DYN and run Z88I1 again. 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*. 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.

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! However, for very large structures you may use Z88 with 64 Bit integers and pointers (i.e. the 64 Bit Z88 versions for Windows Server or Vista 64 bit or LINUX 64-Bit or Mac OS X) to avoid overflows of internal loop counters etc.

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

The Z88 32 Bit versions for Windows and LINUX deal with

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

The Z88 64 Bit versions for Windows and LINUX deal with

- Floating point numbers with Doubles = 8 bytes
- Integers and pointers with Longs = 8 bytes.

However, on several UNIX machines you may compile (compiler switches and compiler directive FR_XQUAD) the solver modules using

- Floating point numbers with long Doubles = 16 bytes = 128 bit
- Integers with long Longs = 8 bytes = 64 bit.

Attention: 64 Bit Integers are usefull for very large structures i.e. > 2 ~ 3 mio. of DOF for avoiding internal overflows. However, using 128 Bit floats is much more time-consuming than 64 Bit floats. Test runs with a SUN FIRE V890 with quad precision at my institute at the University of Bayreuth caused five to ten times more CPU time than double precision! Thus, I recommend using 64 bit integers and 64 bit floats on larger computers.

Critical for the memory are Z88F, Z88I1 and Z88I2. If these modules run, then the rest will run, too. Attention Z88PAR: This solver deals heavily with dynamic memory when running, thus, this solver may run out of memory during operation. If this happens, launch Z88I2 instead.

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 Mesh Generator:

NET START

MAXSE	Maximum number of internal nodes for FE mesh 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.
MAXPR	Maximum number of surface and pressure loads
MAXRBD	Maximum number of boundary conditions (used only by Z88O)
MAXIEZ	For the sparse matrix solver part 1 i.e. Z88I1 only. Z88I1 uses a vector with the size of MAXIEZ. There is no way to pre-determine the needed memory but Z88I1 tells you if MAXIEZ was too small. In this case you must increase MAXIEZ and launch Z88I1 again.
MAXGP	Maximum number of Gauss points (used only by Z88O)

COMMON END

For the Cuthill- McKee program:

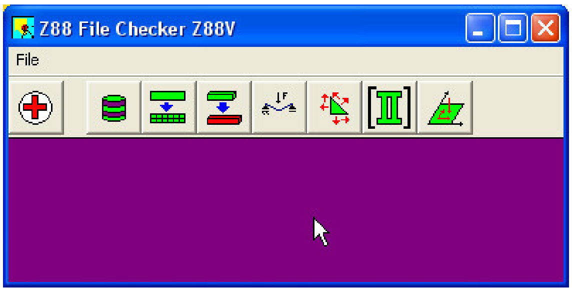
CUTKEE START

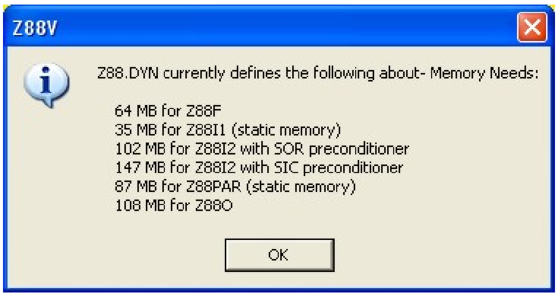
MAXGRA	maximum degree of nodes
MAXNDL	steps of the algorithm

CUTKEE END

DYNAMIC END

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





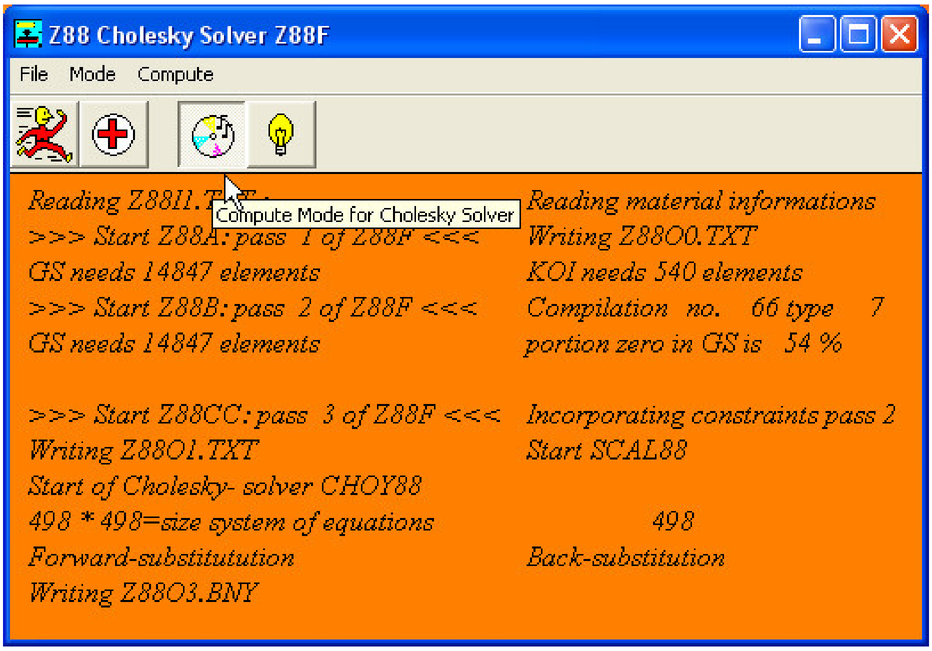
```
Z88.DYN currently defines the following about- Memory Needs:

   64 MB for Z88F
   35 MB for Z88I1 (static memory)
   102 MB for Z88I2 with SOR preconditioner
   147 MB for Z88I2 with SIC preconditioner
   87 MB for Z88PAR (static memory)
   108 MB for Z88O
```

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* structures. For large structures launch the sparse matrix solver Z88I1 and Z88I2.



```
Reading Z88I1.T                        Reading material informations
>>> Start Z88A: pass  1 of Z88F <<<    Writing Z88O0.TXT
GS needs 14847 elements                KOI needs 540 elements
>>> Start Z88B: pass  2 of Z88F <<<    Compilation  no.    66 type     7
GS needs 14847 elements                portion zero in GS is   54 %

>>> Start Z88CC: pass  3 of Z88F <<<   Incorporating constraints pass 2
Writing Z88O1.TXT                      Start SCAL88
Start of Cholesky- solver CHOY88
498 * 498=size system of equations                 498
Forward-substitution                   Back-substitution
Writing Z88O3.BNY
```

Note: The files Z88I1.TXT, Z88I2.TXT and Z88I5.TXT mentioned here are described more precisely in chapter 3.

(1) Compute Mode

Windows: *Z88F > Mode > Compute Mode, Compute > Go*
UNIX: *z88f -c (console) or Z88F -C (Z88COM)*

Input files:

- Z88I1.TXT (general structure data)
- Z88I2.TXT (boundary conditions, constraints)
- Z88I5.TXT (surface and pressure loads), if needed

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) Test Mode

Windows: Z88F > Mode > Test Mode, Compute > Go
UNIX: z88f -t (console) or Z88F -T (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 SPARSE MATRIX SOLVERS

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

The two solver pairs

- Z88I1 and Z88I2 iteration solver
- Z88I1 and Z88PAR direct sparse matrix solver with fill-in

are using only the so-called non-zero elements. They are very suitable for larger structures but need more attention than Z88F. The principal task of every FEA program is the calculation of the displacements. This is the job of the sparse matrix solvers. The calculated deflections are the starting point for a stress calculation with Z88D or nodal force calculation with Z88E.

2.2.1 The Sparse Matrix Iteration Solver Z88I1/Z88I2

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

Sparse Matrix Solver Part 1: Z88I1 is used for both the Iteration Solver Z88I2 and for the direct Sparse Matrix Solver with fill-in Z88PAR. Z88I1 builds the following pointers for the lower part of the total stiffness matrix GS:

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

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)	

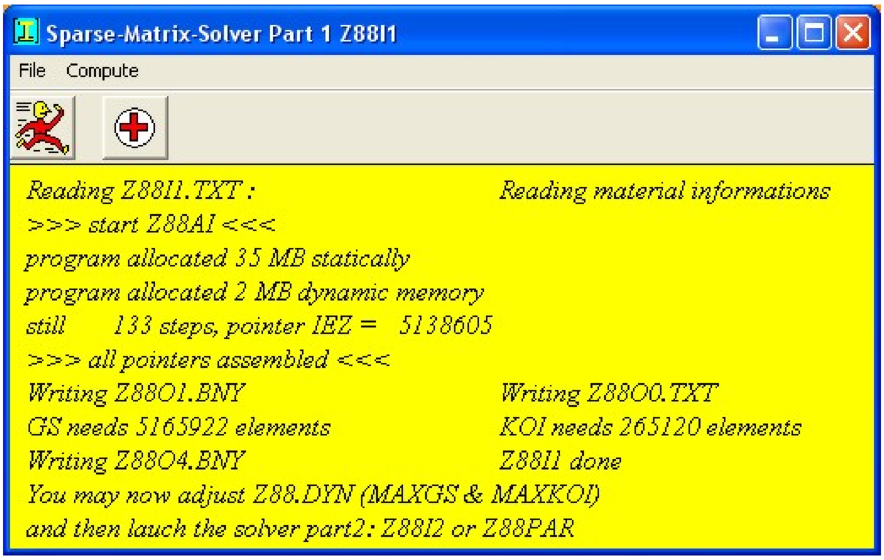
IEZ will result in:

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

and IP:

1	3	5	7	10	13

The pointer IEZ holds MAXIEZ elements, ref. Memory definition file Z88.DYN. You must allocate memory MAXIEZ for the assembly of the sparse matrix. There is no way to predetermine the needed memory but Z88I1 tells you if MAXIEZ was too small. Then, increase MAXIEZ in Z88.DYN and run Z88I1 again. Z88I1 stores the two pointer vectors in a binary file Z88O4.BNY, which may become quite large.



```
Reading Z88I1.TXT :                      Reading material informations
>>> start Z88AI <<<
program allocated 35 MB statically
program allocated 2 MB dynamic memory
still      133 steps, pointer IEZ =    5138605
>>> all pointers assembled <<<
Writing Z88O1.BNY                        Writing Z88O0.TXT
GS needs 5165922 elements                KOI needs 265120 elements
Writing Z88O4.BNY                        Z88I1 done
You may now adjust Z88.DYN (MAXGS & MAXKOI)
and then lauch the solver part2: Z88I2 or Z88PAR
```

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   5200000 ← adjust this before running Z88I2/Z88PAR
    MAXKOI   270000 ← adjust this before running Z88I2/Z88PAR
    MAXK      46000
    MAXE      27000
    MAXNFG   137000
    MAXNEG       32
    MAXIEZ  5200000 ← adjust this before running Z88I1
COMMON END
```

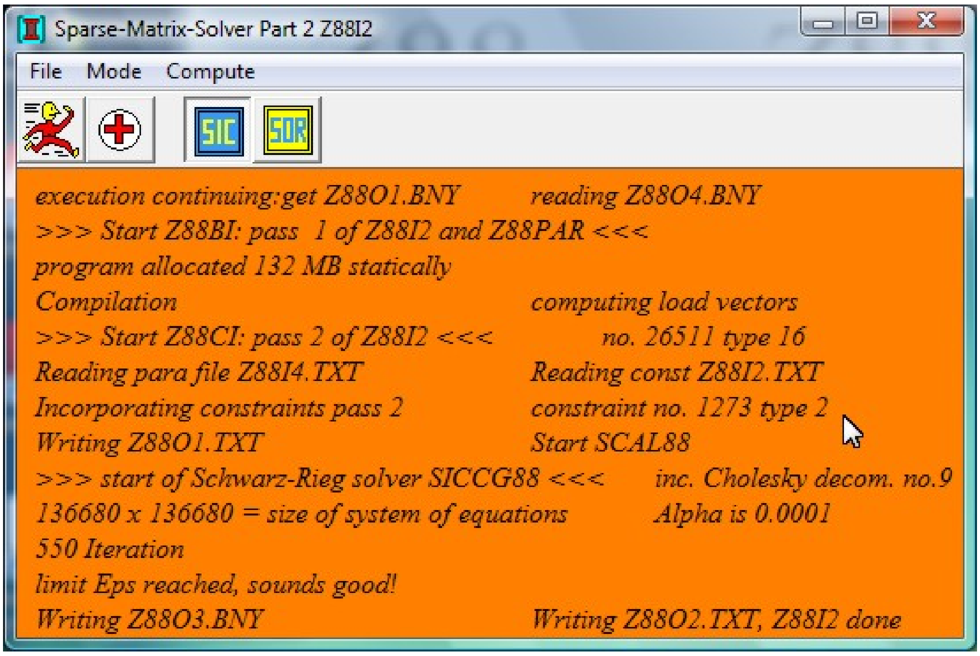
Thus, proceed for large structures for Z88 in 3 or more steps:

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 MAXIEZ increase MAXIEZ in Z88.DYN and run Z88I1 again. Repeat this step until Z88I1 completes properly.

Sparse Matrix Iteration Solver Part 2: Z88I2 computes the element stiffness 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 favourite pre-conditioner: Either a *SOR* step or a so-called incomplete Cholesky decomposition (*shiftet incomplete Cholesky decomposition SIC*). Default is *SIC* preconditioning because the main parameter, the so-called shift factor α is easy to handle. The *SOR* preconditioning needs only ~ 2/3 of the memory of SIC but the SOR parameter ω cannot be determined *a-priori*.



```
execution continuing:get Z88O1.BNY        reading Z88O4.BNY
>>> Start Z88BI: pass  1 of Z88I2 and Z88PAR <<<
program allocated 132 MB statically
Compilation                               computing load vectors
>>> Start Z88CI: pass 2 of Z88I2 <<<          no. 26511 type 16
Reading para file Z88I4.TXT               Reading const Z88I2.TXT
Incorporating constraints pass 2          constraint no. 1273 type 2
Writing Z88O1.TXT                         Start SCAL88
>>> start of Schwarz-Rieg solver SICCG88 <<<   inc. Cholesky decom. no.9
136680 x 136680 = size of system of equations  Alpha is 0.0001
550 Iteration
limit Eps reached, sounds good!
Writing Z88O3.BNY                         Writing Z88O2.TXT, Z88I2 done
```

(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 5 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 SIC convergence acceleration. Shift factor *Alpha* (from 0 to 1, good values may vary from 0.0001 to 0.1). For further information consult the literature)
- parameter for the SOR convergence acceleration. Relaxation factor *Omega* (from 0 to 2, good values may vary from 0.8 to 1.2).
- number of CPUs (Z88PAR only, max. 9)

Note: The files Z88I1.TXT, Z88I2, Z88I4.TXT and Z88I5.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 sparse matrix solvers part 2: Z88I2 and Z88PAR)
- Z88I5.TXT (surface and pressure loads), if needed

Output files:

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

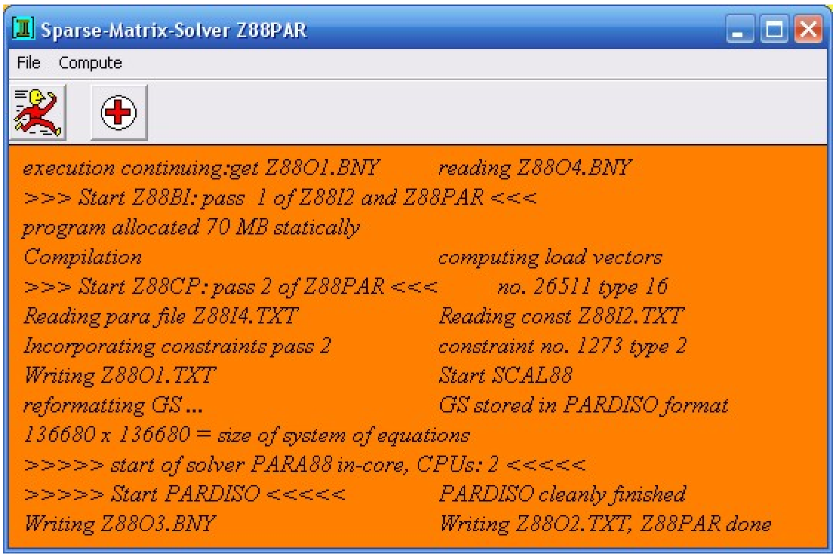
In addition two binary files Z88O1.BNY and Z88O3.BNY are generated. These binary files are later used by Z88D (stress processor) and Z88E (nodal force processor).

2.2.2 The direct Sparse Matrix Solver with Fill-In Z88PAR

Before running Z88PAR launch the firts solver part Z88I1, see Chp. 2.2.1.

Sparse Matrix Solver Part 2: Z88PAR. The solver Z88PAR does a matrix decomposition but in contrary to simple Z88F this solver operates with fill-in. Fill-in means allocating dynamic memory for the new matrix elements created by the decomposition process. Thus, the memory needs cannot calculated before starting Z88PAR. If the memory exhausts while Z88PAR is running, Z88PAR will quit with an error message.

This solver works at very high speed for not too large structures (100.000 ~ 1.000.000 DOF) because it can use up to 9 CPUs, but needs tremendous more memory than the sparse matrix iteration solver Z88I2. Thus, Z88PAR is only really usefull on machines with very much memory and 64 bit pointers and integers. I recommend for Z88PAR the 64 bit version of Z88, a 64 bit Windows operating system and a minimum of 4 GByte (8 or 16 Gbyte is better) of memory. When using a 32 bit operating system and 4 GByte of memory you are limited to ~ 150.000 DOF with Z88PAR. The solver core used is PARDISO by O. Schenk, University of Basel, Switzerland. Because of legal reasons we cannot publish the sources of Z88PAR – the one exception of our Open Source policy.



```
execution continuing:get Z88O1.BNY          reading Z88O4.BNY
>>> Start Z88BI: pass  1 of Z88I2 and Z88PAR <<<
program allocated 70 MB statically
Compilation                                 computing load vectors
>>> Start Z88CP: pass 2 of Z88PAR <<<           no. 26511 type 16
Reading para file Z88I4.TXT                 Reading const Z88I2.TXT
Incorporating constraints pass 2            constraint no. 1273 type 2
Writing Z88O1.TXT                           Start SCAL88
reformatting GS ...                         GS stored in PARDISO format
136680 x 136680 = size of system of equations
>>>>> start of solver PARA88 in-core, CPUs: 2 <<<<<
>>>>> Start PARDISO <<<<<                   PARDISO cleanly finished
Writing Z88O3.BNY                           Writing Z88O2.TXT, Z88PAR done
```

Enter the number of CPUs you want to use by the 5th entry in Z88I4.TXT with a maximum of 9 CPUs. The first four entries have no meaning for Z88PAR but must exist. Pay attention *not* to set these variables in the Windows environment, i.e. in the system settings:

NUM_THREADS, OMP_SET_NUM_THREADS.

Otherwise, this may conflict with entries Z88I4.TXT. For the input and output files see 2.2.1.

2.2.3 Which solver should i use?

Roughly spoken: Use the simple and reliable Cholesky solver Z88F for small structures. The sparse matrix iteration solver Z88I1/Z88I2 works *always* even for very large structures under 32 bit operating systems. For medium sized structures the direct sparse matrix solver with fill-in Z88I1/Z88PAR is very suitable because of its tremendous speed.

Solver	Type	Number of DOF	Memory needs	Speed	Multi-CPU	Notes
Z88F	Cholesky solver without Fill-In	up to ~ 30.000	medium	medium	no	running Z88H before Z88F is recommended
Z88I1/ Z88PAR	direct Solver with fill-in	up to~ 150.000 for 32 bit PCs	very high	very high	yes	usefull with several CPUs and *very much* memory
Z88I1/ Z88I2	conjugated gradients solver with pre-conditioning	no limits (up to 5 mio. DOF were run on an ordinary PC)	an absolute minimum	medium	no	a very stable and reliable solver for very large structures

2.3 THE STRESS PROCESSOR Z88D

A stress calculation with Z88D can run only if the deflections were calculated by Z88F or Z88I1/Z88I2 or Z88I1/Z88PAR 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/Z88I2 or Z88I1/Z88PAR 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:

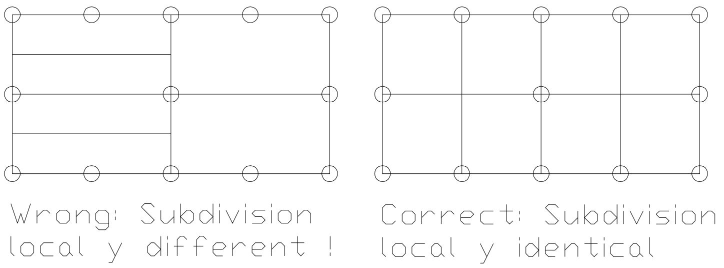
- Local x axis runs in direction of local nodes 1 and 2
- Local y axis runs in direction of local nodes 1 and 4
- Local z axis runs in direction of local nodes 1 and 5

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 axes can be subdivided as follows:

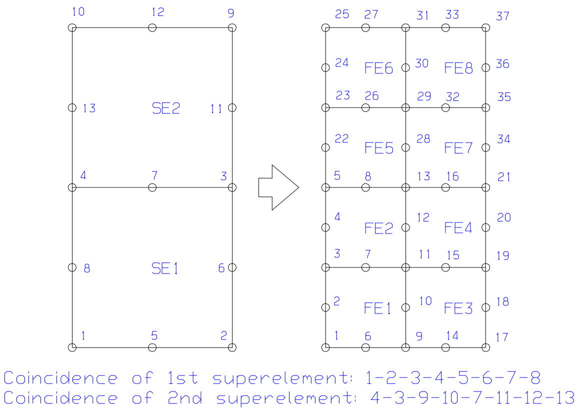
- Equidistant
- Increasing geometrically from node 1 to 4 or 5: Mesh becomes rougher
- Decreasing geometrically from node 1 to 4 or 5: Mesh becomes finer

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 axes 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):



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 axes 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 (structure data) has always cartesian coordinates and KFLAG is set to 0.

2.6 THE OPENGL PLOT PROGRAM Z88O

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. 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 by *Shift-Print* into the clipboard and proceed with Windows' program Paint or Corel Paint.

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. On LINUX systems you should install the genuine NVIDIA graphics driver from www.nvidia.com if you've got a NVIDIA graphics card – the speed will increase heavily in contrast to the standard LIXUX graphics driver.

Of course, you may define your choice of colors, the light features, material properties, the polygon offset in the parameter files Z88O.OGL (for Windows) and Z88.FCD (for LINUX/UNIX and Mac OS X). Be careful with changes in Z88O.OGL (Windows) or Z88.FCD (UNIX/LINUX and Mac OS X): 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 cannot 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 with the very first *Run* pushbutton.

Needed Files:	Super Struktures	undeflected FE Struktures	deflected FE Struktures
Z88NI.TXT	yes	no	no
Z88I1.TXT	no	yes	yes
Z88I2.TXT	no	yes for displaying the boundary conditions	yes for displaying the boundary conditions
Z88I5.TXT	no	yes for displaying the surface and pressure loads	yes for displaying the surface and pressure loads
Z88O2.TXT	no	no	yes
Z88O5.TXT	no	yes for displaying the stresses in the Gauss points	no
Z88O8.TXT	no	yes for displaying the stresses in the corner nodes or the average element stresses	yes for displaying the stresses in the corner nodes or the average element stresses

Z88 deals with these files

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/ENGINEER 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 or Z88I1 with Z88PAR) 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 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 stress concentration factor which is impossible for a FEA system which deals with a *whole structure of beams*. Of couse, you may compute the stresses in a chamfer by putting a FE mesh around it. But this needs either plane stress elements or 3D elements but neither beam elements nor cam elements.

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. However, this is odd for OpenGL in some modes so i decided for the following way after a lot of experiments:

- *von Mises/principla/Tresca 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 in the Gauss points, 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/principla/Tresca 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/principla/Tresca 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.

- *von Mises/principla/Tresca stresses in Gauss points.* This is most accurate but delivers not always very pretty pictures. INTORD must be greater than 0.

Z88O may show these stresses after a run of Z88D – but only one type of stress:

- *von Mises stresses*
- *Rankine or principal stresses*
- *Tresca stresses*

Thus, if you have computed the *von Mises* stresses with a Z88D run Z88O will show the *von Mises* stresses. If you want to see now the *Tresca* stresses you must leave Z88O. Edit Z88I3.TXT, enter the proper parameter (in this case, set the third entry to 3) and re-run Z88D. Then start Z88O again. This looks awkward but don't you know before starting the FE computations which type of stresses is suitable and correct for your task?

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. You may plot the shaded colors for stresses or for the deflections or the hidden line display or the wire frame display with the deflected structure. The display of the stresses in Gauss points in only possible for undeflected structures.

	3D	2D	BC	undef.	deflec.	nodes	elem.
Light	+	+	+	+	+	-	-
Hidden Line	+	-	+	+	+	o	-
Wire Frame	+	+	+	+	+	+	+
Stresses in corner nodes	+	+	-	+	+	-	-
Stresses aver. elements	+	+	-	+	+	-	-
Stresses in Gauss points	+	+	-	+	-	-	+
Deflections X	+	+	-	+	+	-	-
Deflections Y	+	+	-	+	+	-	-
Deflections Z	+	+	-	+	+	-	-

combination of the different modes of Z88O

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 you may use mouse navigation: Under Windows, press the mouse icon. Under UNIX, press the pushbutton *Keyboard/ 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, LINUX and Mac OS X:

Key Windows & LINUX	Key Mac OS X	Function
PRIOR	fn + ↑	increase zoom
NEXT	fn + ↓	decrease zoom
CURSOR LEFT	←	panning X
CURSOR RIGHT	→	panning X
CURSOR UP	↑	panning Y
CURSOR DOWN	↓	panning Y
HOME		panning Z
END		panning Z
F2	fn + F2	rotate around X axis
F3	fn + F3	rotate around X axis
F4	fn + F4	rotate around Y axis
F5	fn + F5	rotate around Y axis
F6	fn + F6	rotate around Z axis
F7	fn + F7	rotate around Z axis
F8	fn + F8	reset all rotations to 0

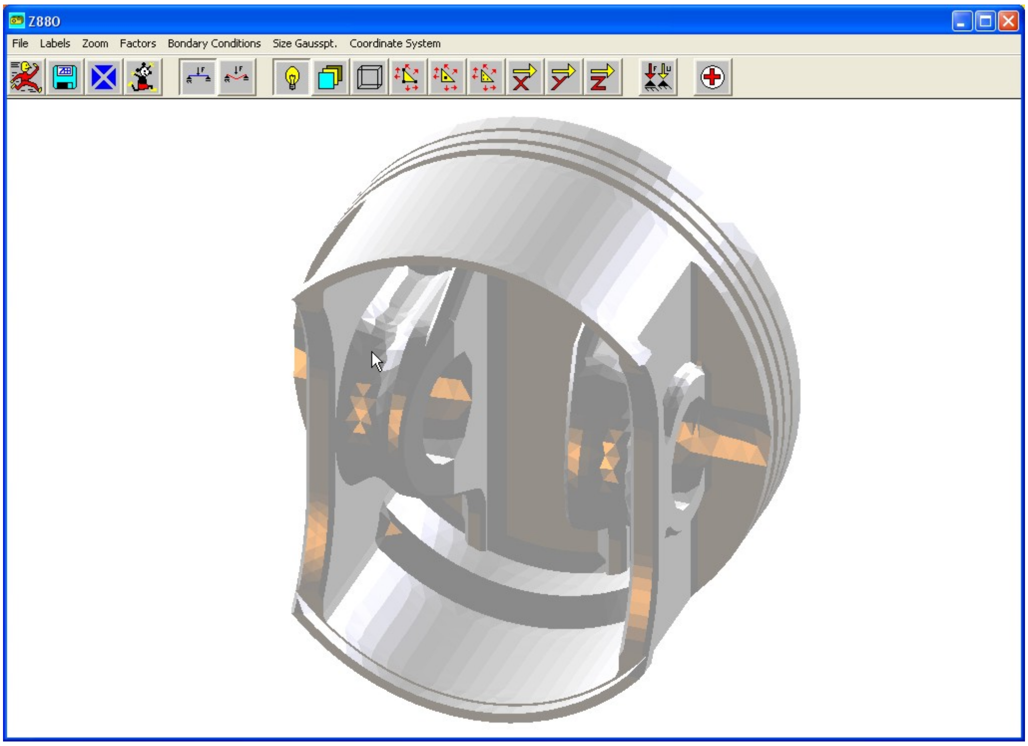
Special hints for MacBook:

Start Z88COM. Goto X11 > Preferences > Input: activate „emulate three button mouse“

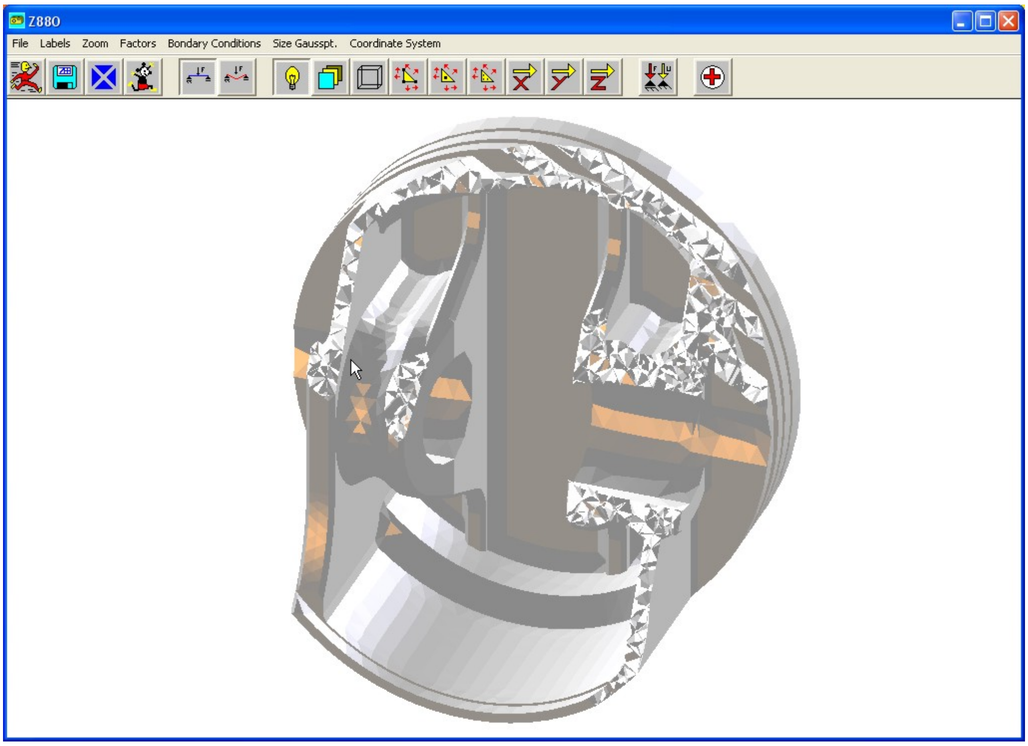
- Zoom in/out: move with touchpad pressed
- Panning: *alt* + move with touchpad pressed
- Rotating: *cmd* + move with touchpad pressed

USB-plugged three button mice operate just like the Windows version, see manual

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: use *Factors > Z limit towards you*. 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.

The menu items of Z88O:

Name of Structure File: Windows: the diskette icon. UNIX: the *File* pushbutton
Choose the structure file here. Enter name, if necessary with path. The new stucture is loaded. You'll start rendering with the icon *Go* or with the *Run* pushbutton. This mode exists only for a first entry control of an undeflected structure. Please keep in mind: To use *all* display modes, Z88O needs the files Z88I1.TXT (structure data), Z88I2.TXT (boundary conditions), Z88I5.TXT (surface and pressure loads, if given), Z88O2.TXT (the computed deflections), Z88O5.TXT (stresses from Z88D) and Z88O8.TXT (stresses from Z88D).

Deformation Modes of the Structure: the proper icons or pushbuttons
Plots the undeflected structure or the deflected structure. You may do all other rendering operations with the undeflected structure or the deflected structure. Exception: displaying the Gauss stresses in a *deflected* structure is not possible.
Caution Deflected: The user must have executed a calculation of displacements before using this function. Do a FEA run with Z88F or Z88I1/Z88I2 or Z88I1/Z88PAR before using Z88O. Otherwise, some old files Z88O2.TXT (displacements) from earlier Z88 runs are opened causing totally wrong results!

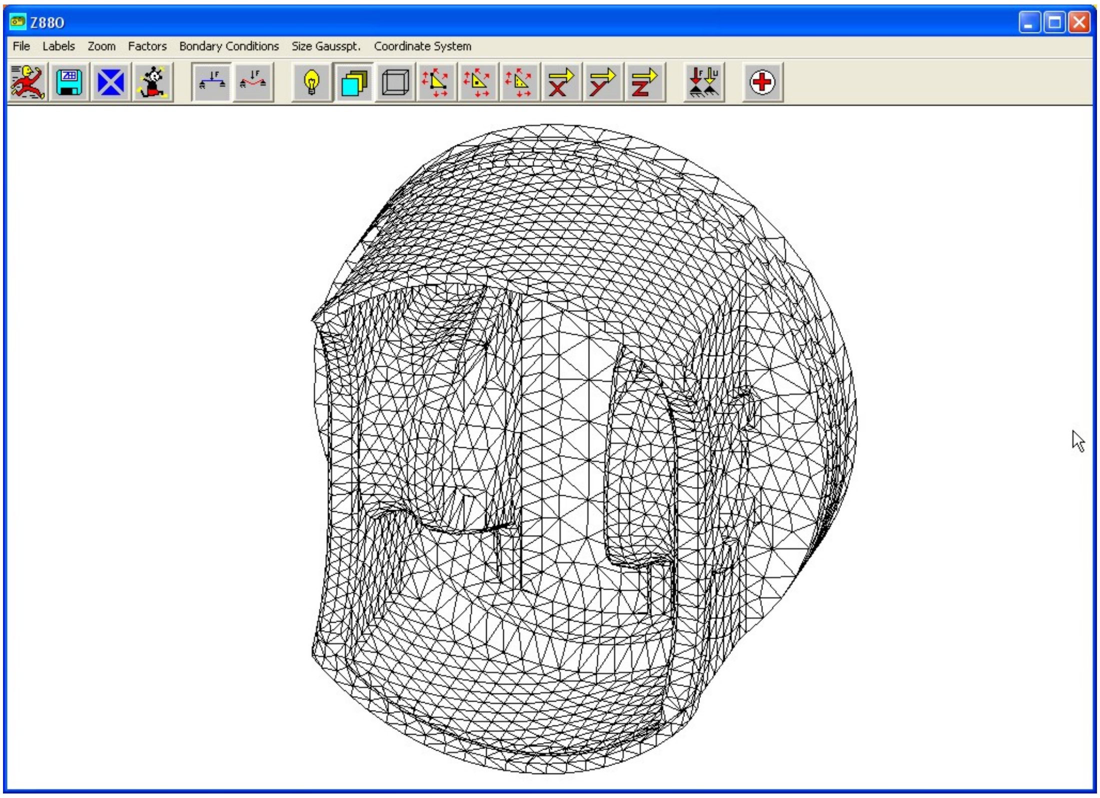
Choice of the 3D effects: the proper icons or pushbuttons

1. **Light on**. 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. **Hidden Lines**. 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). In this mode you cannot see *all* desired nodal and elements labels because some labels are hidden. The polygon offset can be edited in the header files Z88O.OGL (Windows) and Z88.FCD (UNIX).
3. **Wire Frame**. *All* lines are plotted, thus, this is the proper mode for 2D struktures and very simple 3D structures. Only in this mode you can see *all* desired nodal and elements labels.
4. The ***von Mises/principla/Tresca* stresses of the corner nodes** are plotted. In fact, the

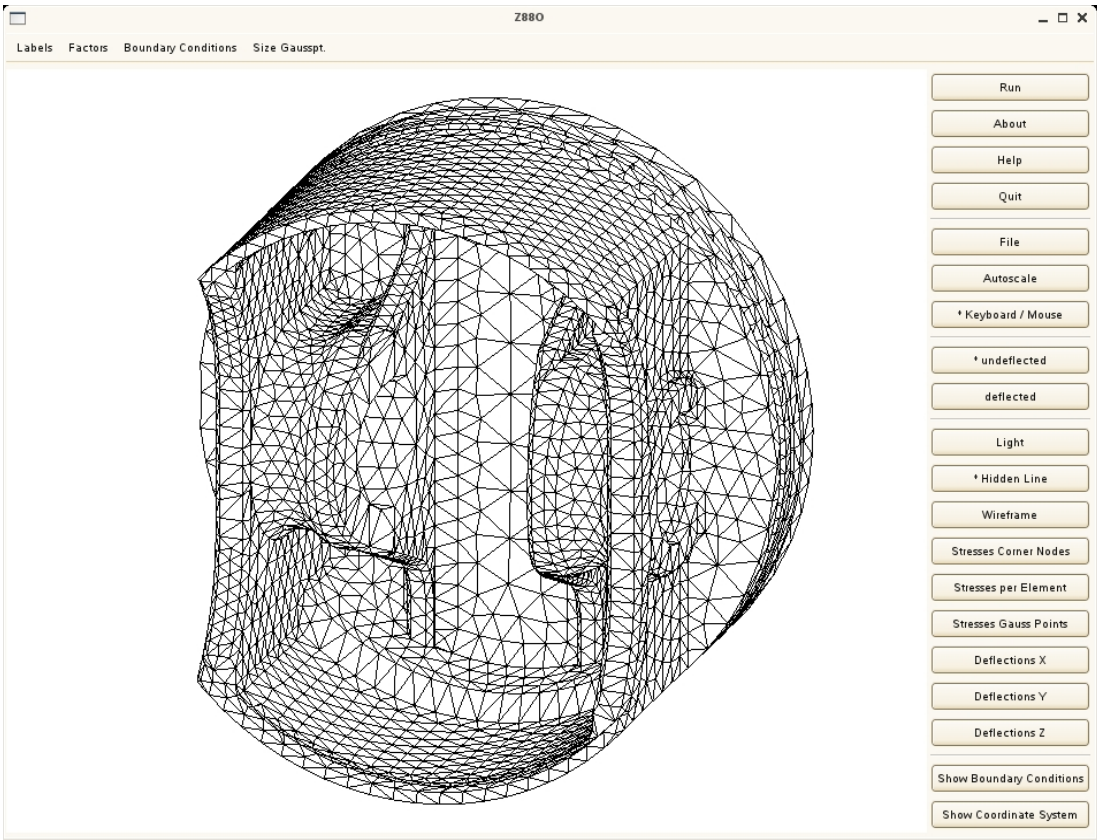
stresses are computed not really in the corner nodes which would lead to very wrong results especially for very tapered elements but in Gauss points laying in the near of the current corner nodes. Stresses are computed for just the same number of Gauss points like the number of corner points. Because 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 in Gauss points, 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.

5. The ***von Mises/principal/Tresca* 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/principal/Tresca 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.
6. The ***von Mises/principal/Tresca* stresses in the Gauss points** are plotted. This is the most accurate mode but leads not always to very nice pictures. You may change the size of the Gauss points in the menu. The value of the order of integration INTORD in the header file Z88I3.TXT is important and INTORD must be greater than 0.
7. Plot of the displacements for X with color shading
8. Plot of the displacements for Y with color shading
9. Plot of the displacements for Z with color shading

For pos. 4. to 9. the color range may be edited in the header files Z88O.OGL (Windows) and Z88.FCD (UNIX).



Windows: Hidden line plot of the BMW piston.



LINUX and Mac OS X: Hidden line plot of the BMW piston.

Drawing Node and Element Numbers: *Labels > No Labels, Nodes, Elements, Label all*
Plot the element numbers or the node numbers or skip numbering. 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*. In *Label all* mode the element and node numbers you have chosen before are plotted. If you want to see all numbers again but you have forgotten how many nodes and elements are in your structure you may enter a very high number e.g. *from 1 to 10000000*. Z88O computes then the exact number. Please remember that you'll only get rendered all desired labels on the surfaces if you are in *Wire Frame* mode. The other modes may hide some labels. And labels inside a structure are usually covered by the tetrahedron and hexahedron surfaces.

Zooming:
Keyboard: *PRIOR* and *NEXT*
mouse navigation on: *left mouse button pressed*

Panning:
Keyboard: X: *CURSOR LEFT* and *CURSOR RIGHT*
Y: CURSOR UP and *CURSOR DOWN*
Z: HOME and *END*
mouse navigation on: *middle mouse button pressed*

Rotating:
Keyboard: F2~F7: rotate in 10° steps. F8 resets to 0.
Menu: *Faktors > Rotations 3D*
mouse navigation on: *right mouse button pressed*

Enlarging Deflections:
Menu: *Factors > Deflections > FUX, FUY, FUZ*

Some remarks on stresses:
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/principal/Tresca* stresses either in the corner nodes or as mean values per each element or in the Gauss points. And before running the stress processor Z88D you really had to calculate the displacements by running Z88F or one of the sparse matrix solvers. Thus, the sequence is:

1. run Cholesky solver Z88F or sparse solvers Z88I1/Z88I2 or Z88I1/Z88PAR
2. Edit the parameter file Z88I3.TXT. Mind the third entry: 1= *von Mises* stresses, 2= *principal* stresses, 3=*Tresca* stresses.
3. run stress processor Z88D
4. run Z88O, if you want to plot stresses

Caution: The operator is responsible for first running a stress calculation by Z88D before using this function. Otherwise some old stress files Z88O5.TXT and Z88O8.TXT from earlier calculations are read in causing totally wrong results!

Automatic Scaling: the appropriate icon or pushbutton
The Autoscale function takes care that structures will completely fit on the screen. Autoscale activates automatically if a new structure is loaded.

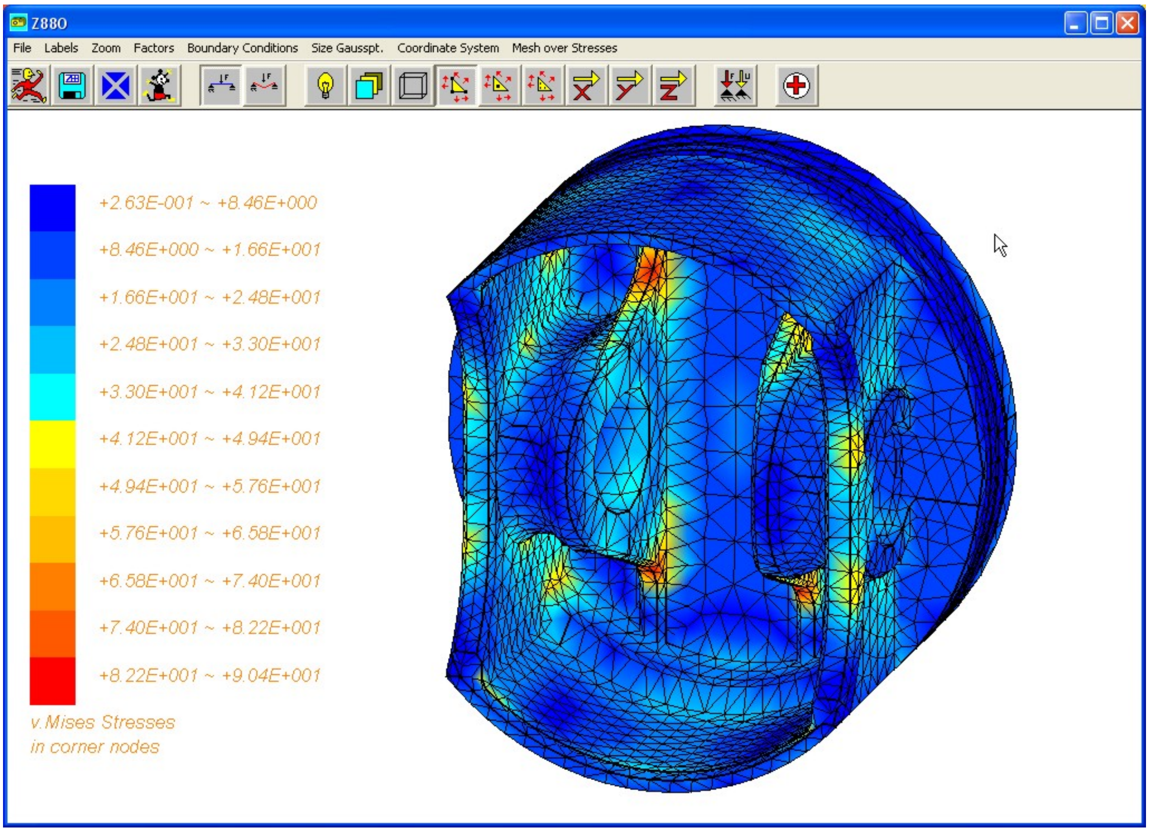
Menu item **Boundary Conditions**:
You may choose which boundary conditions to show. Sometimes you may want to hide some BCs for a better overview. You may also change the size of the BCs points.

Menu item **Size Gausspts.**:
You may change the size of the Gauss points to get a nice picture.

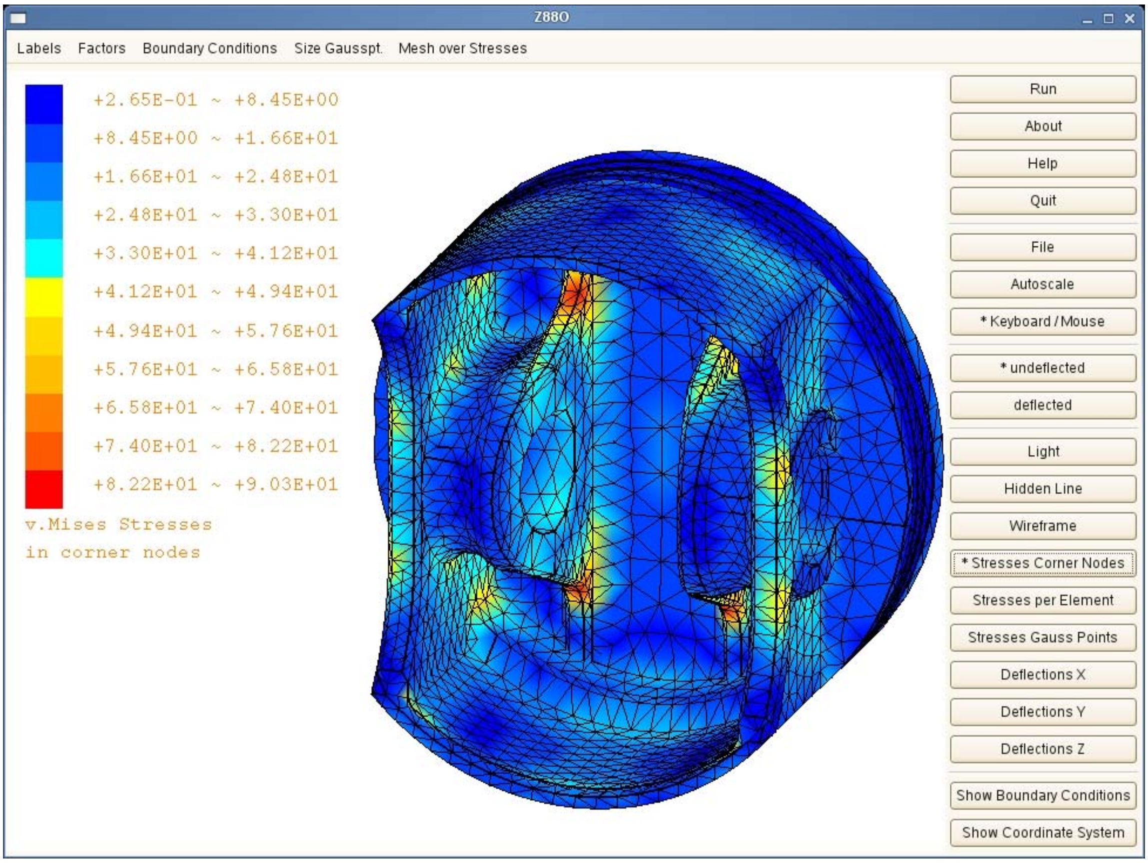
Menu item **Coordinate System**:
You may switch on or off the coordinate system. Default is on.

Menu item **Mesh over Stresses**:
For 3D structures you may switch on or off the mesh i.e. the hidden line display over the stresses display. Default is on. For very large structures you should switch off *Mesh over Stresses* because this means heavy computing load.

Height Ratio FYCOR: Files Z88O.OGL and 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/LINUX and Mac OS X). Load a perfectly circular or perfectly square structure and modify FYCOR until 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. Mesh over Stresses is switched on.



LINUX and Mac OS X: plot of the von Mises stresses in the corner nodes of the BMW piston. Mesh over Stresses is switched on.

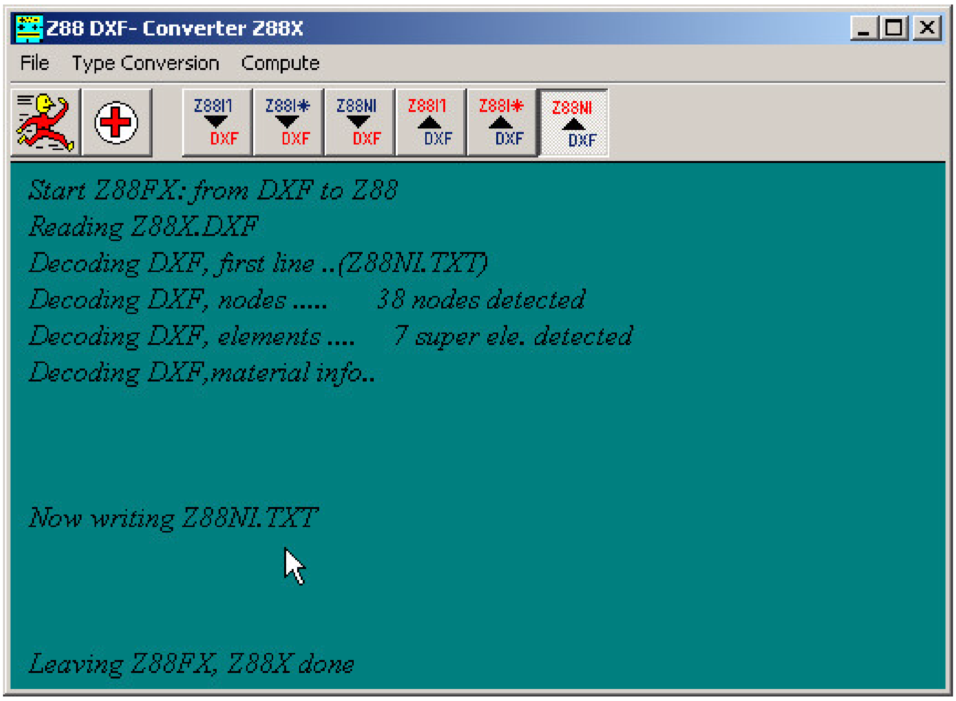
2.7 THE CAD CONVERTER Z88X

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



```
Start Z88FX: from DXF to Z88
Reading Z88X.DXF
Decoding DXF, first line ..(Z88NI.TXT)
Decoding DXF, nodes .....      38 nodes detected
Decoding DXF, elements ....      7 super ele. detected
Decoding DXF,material info..


Now writing Z88NI.TXT


Leaving Z88FX, Z88X done
```

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 V12 has been intensively tested together with the different AutoCAD and AutoCAD LT versions for Windows of Autodesk, and AutoDesk's DXF guidelines have been regarded as the inventor of the DXF interface, according to AC1009 and AC1012. **Choose AutoCAD R12 DXF format, if in doubt!**

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) Define the surface and pressure loads (if needed) on the layer Z88FLA.

(9) 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, Z88I3.TXT and Z88I5.TXT (if needed)*

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 Z88O*
- *Direct Cholesky Solver Z88F*
- *Sparse Matrix Solvers Z88I1/Z88I2 or Z88I1/Z88PAR*
- *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, Z88I3.TXT and Z88I3.TXT (if needed)*

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.
Z88FLA: Layer containing the surface and pressure loads as defined for Z88I5.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 centre 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.

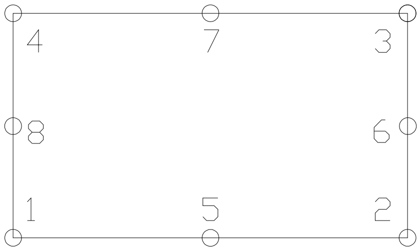
Then you do the same with the second element. Remember: **You determine with this order which of the elements will be the real second element now**. In the previous 4th step you have only defined what kind of element the second element is. You determine here **how** the element is defined topologically.

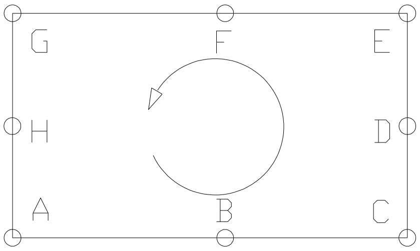
The third element follows and so on. If you should make a mistake at the outlining of an element then delete all previous lines of this element (e.g. with an UNDO function) and start again at the first point of the questionable element. But if you notice now just outlining element 17 that you have made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

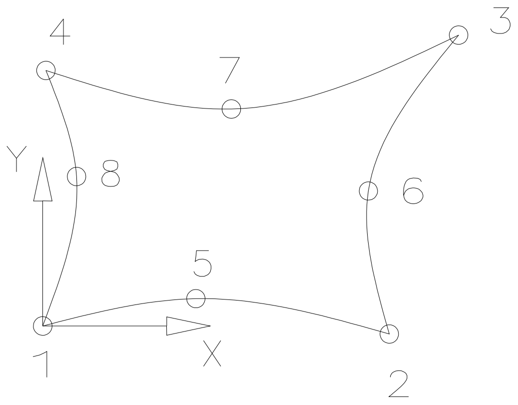
First the corner nodes, then the middle nodes, reads 1-2-3-4-5-6-7-8. The coincidence list must look like this in the Z88 input files. However, for Z88X' use for comfortably outlining the elements the order is 1-5-2-6-3-7-4-8-1 (left picture) respectively A-B-C-D-E-F-G-H-A (right picture):



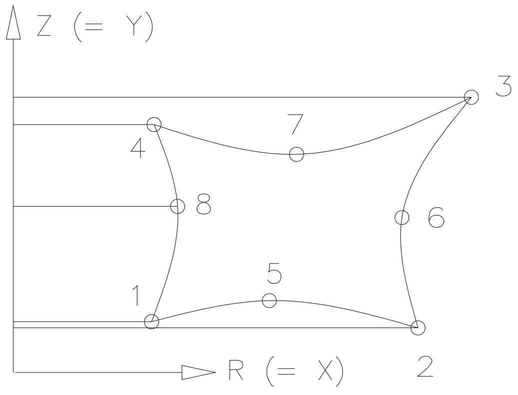


Following the CAD outline orders for all elements but No. 16 and No.17 (because these tetrahedrons can only machine- generated, nearly impossible by hand):

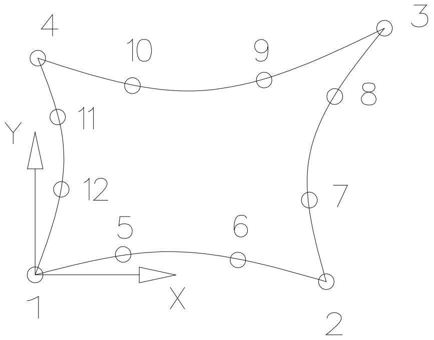
<u>Element No.7 and No.20:</u> 1 - 5 - 2 - 6 - 3 - 7 - 4 - 8 - 1



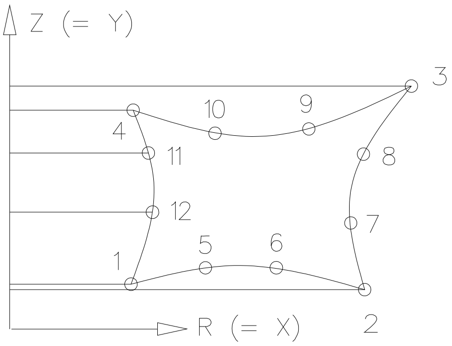
<u>Element No.8:</u> 1 - 5 - 2 - 6 - 3 - 7 - 4 - 8 - 1



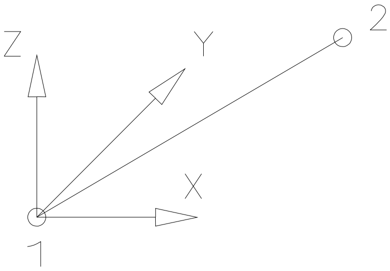
<u>Element No.11:</u> 1 - 5 - 6 - 2 - 7 - 8 - 3 - 9 - 10 - 4 - 11 - 12 - 1



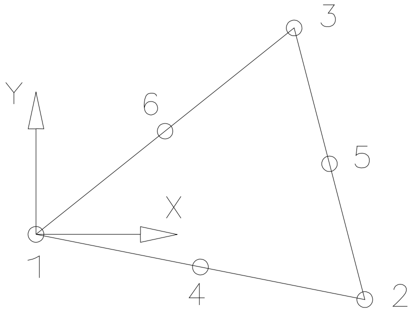
Element No.12: **1 - 5 - 6 - 2 - 7 - 8 - 3 - 9 - 10 - 4 - 11 - 12 - 1**



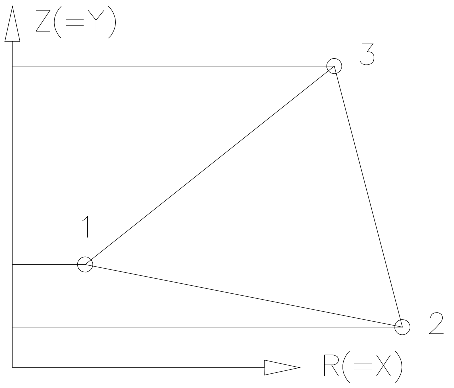
Element No. 2, 4, 5, 9, 13: **Line from node 1 to node 2**



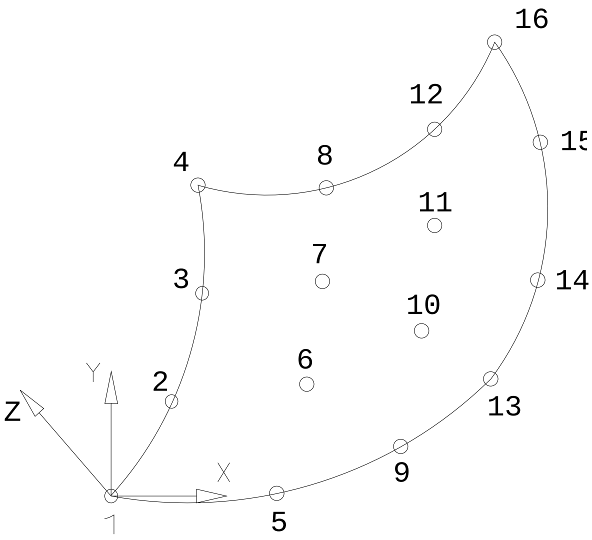
Element No.3, 14, 15 and 18: **1 - 4 - 2 - 5 - 3 - 6 - 1**

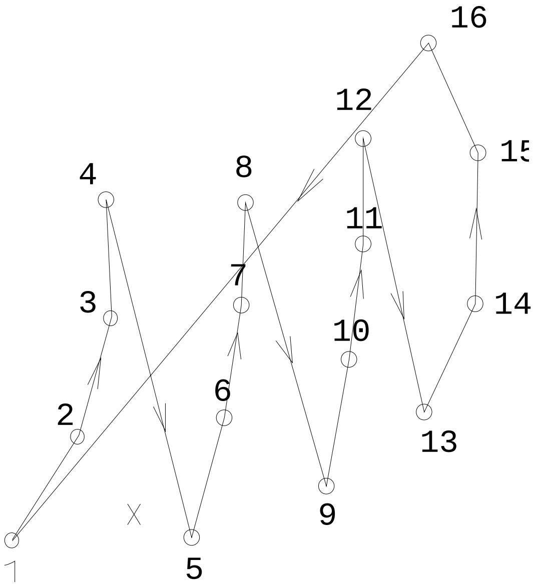


Element No.6: **1 - 2 - 3 - 1**

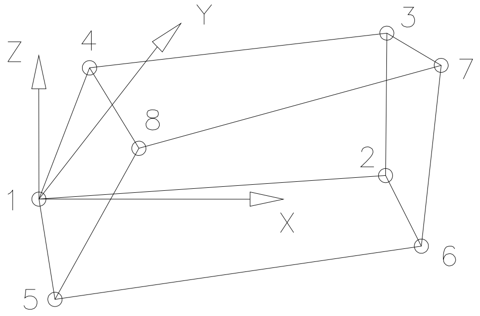


Element No.19: **1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 -10 - 11 - 12 - 13 - 14 - 15 - 16 - 1**



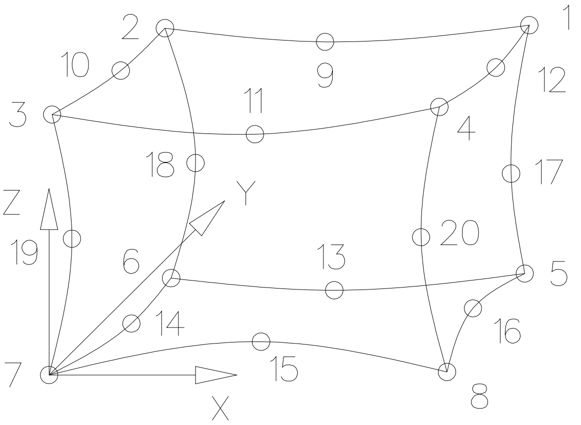


Element No.1:
Upper plane: 1 - 2 - 3 - 4 - 1, quit LINE function
Lower plane: 5 - 6 -7 - 8 - 5, quit LINE function
1 - 5, quit LINE function
2 - 6, quit LINE function
3 - 7, quit LINE function
4 - 8, quit LINE function



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)
Surface and pressure loads 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 0 1*

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)
Surface and pressure loads 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 0

6.2 Material information lines:

For every material information one separate line:

MAT
Number of the material information
This material information starts with <u>*element no. abc*</u> *inclusively*
This material information ends with <u>*element no. xyz*</u> *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)

<u>*... And if beams (but not plates !) are defined in addition:*</u>

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)

<u>*... And if plates (but not beams !) are defined in addition:*</u>

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: if surface and pressure loads are defined: create the layer **Z88FLA** and activate it. Write with the TEXT function into a free space (well into any place of your drawing):

8.1 Number of surface and pressure loads

i.e. the first input group of the surface and pressure loads file Z88I5.TXT

Z88I5.TXT number of surface and pressure loads

Write into one line, separate each item by at least one blank. **Make sure to write in the layer Z88FLA.**
Example: The structure features 12 surface loads. Thus: *Z88I5.TXT 12*

8.2 Surface and pressure loads

i.e. the second input group of the surface and pressure loads file Z88I5.TXT

FLA number of the surface and pressure load

<u>The following entries depend from the element type with surface and pressure load:</u>

→ Plain stress element No.7 and 14 and Torus elements No.8 and 15:
Element number with surface load
Pressure, positive if poiting towards the edge
Tangential shear, positive in local r-direction
3 nodes of the loaded edge

Example: The plain stress element 97 is the third element with surface load. The load should be applied onto the edge defined by the corner nodes 5 and 13 and by the mid node 51. One surface load is applied normally to the edge with 100 N/mm and the other surface load is applied tangentially and positive in local r direction with 300 N/mm (defined by the two corner nodes). Thus: *FLA 3 97 100. 300. 5 13 51*

→ Hexahedron No.1:
Element number with surface and pressure load
Pressure, positive if poiting towards the surface
Tangential shear, positive in local r direction
Tangential shear, positive in local s direction
4 nodes of the loaded surface

Example: The hexahedron 356 is the 34th element with surface loads. The load should be applied onto the surface defined by the corner nodes 51, 34, 99 und 12 .The first surface load is pressure with 100 N/mm. The second surface load is applied tangentially and positive in local r direction with 200 N/mm. The third surface load is applied tangentially and positive in local s direction with 300 N/mm . Thus
FLA 34 356 100. 200. 300. 51 34 99 12

→ Hexahedron No.10:
Element number with surface and pressure load

Pressure, positive if poiting towards the surface
Tangential shear, positive in local r direction
Tangential shear, positive in local s direction
4 nodes of the loaded surface

→ **Plate elements No.18, 19 and 20:**
Element number with pressure load
Pressure, positive if poiting towards the surface
(It is easier to enter the pressure loads for plate elements directly into Z88I1.TXT than via Z88I5.TXT)

Separate each item by at least one blank. **Make sure to write in the layer Z88FLA.**

9th 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 3D 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*. For plates the direct entry of the pressure load is allowed. When using *Wildfire 2*, do not forget to define an analysis. Otherwise, no boundary conditions are filed!

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.

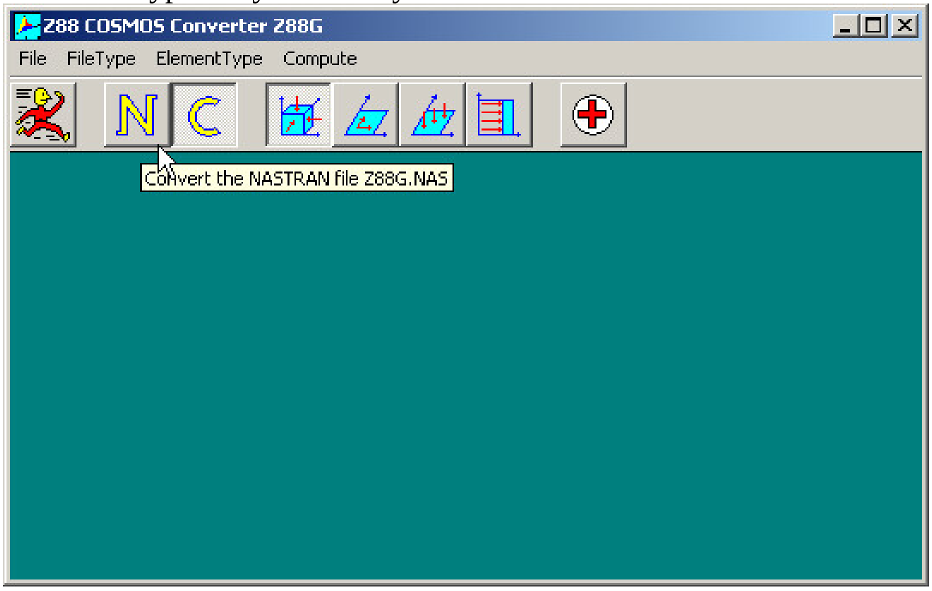
Then lauch the converter **Z88G**. The converter produces the Z88 input files Z88I1.TXT, Z88I2.TXT and Z88I3.TXT and Z88I5.TXT (if needed) automatically. You may then enter the Z88 input files and edit values e.g. material data and integration orders, if necessary.

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.
Keep in mind that those exchange file formats and their Pro/ENGINEER output are subject to change every some months. Visit www.z88.de or www.z88.org for updated versions of Z88G.

You may create the following Z88 element types with Z88G:

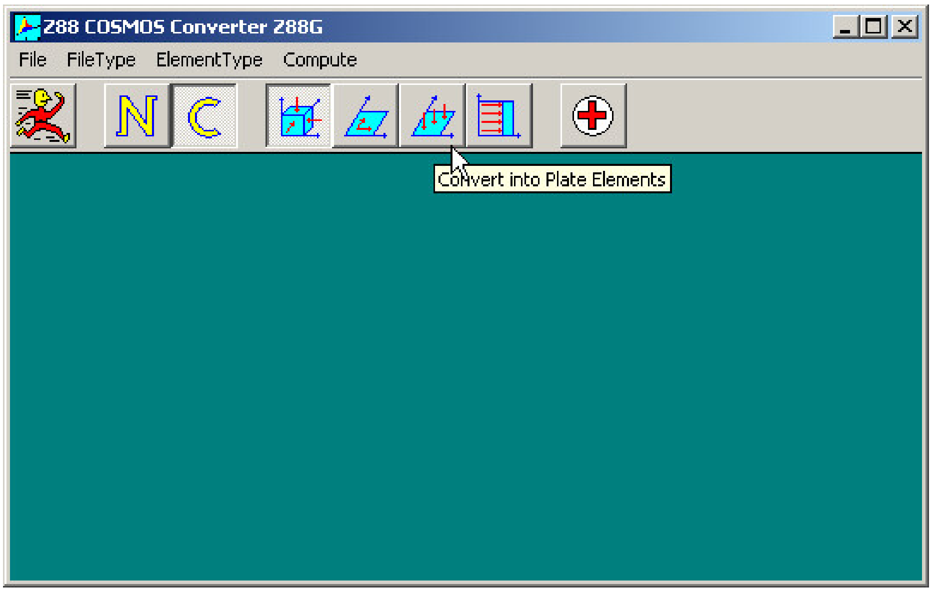
- Tetrahedron No.16 (*Tetrahedron*, *parabolic* in Pro/ENGINEER)
- Tetrahedron No.17 (*Tetrahedron*, *linear* in Pro/ENGINEER)
- Plane stress No.14 (*Shell*, *triangle*, *parabolic* in Pro/ENGINEER)
- Plane stress No.7 (*Shell*,*quadrangle*, *parabolic* in Pro/ENGINEER)
- Plate No.18 (*Shell*, *triangle*,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

Please keep in mind that Z88G is capable to deal directly with pressure loads from Pro/ENGINEER only with NASTRAN files. In this case, the file for surface and pressure loads Z88I5.TXT is generated. This is not possible for COSMOS files: Here you are to enter pressure loads via nodal forces.
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.



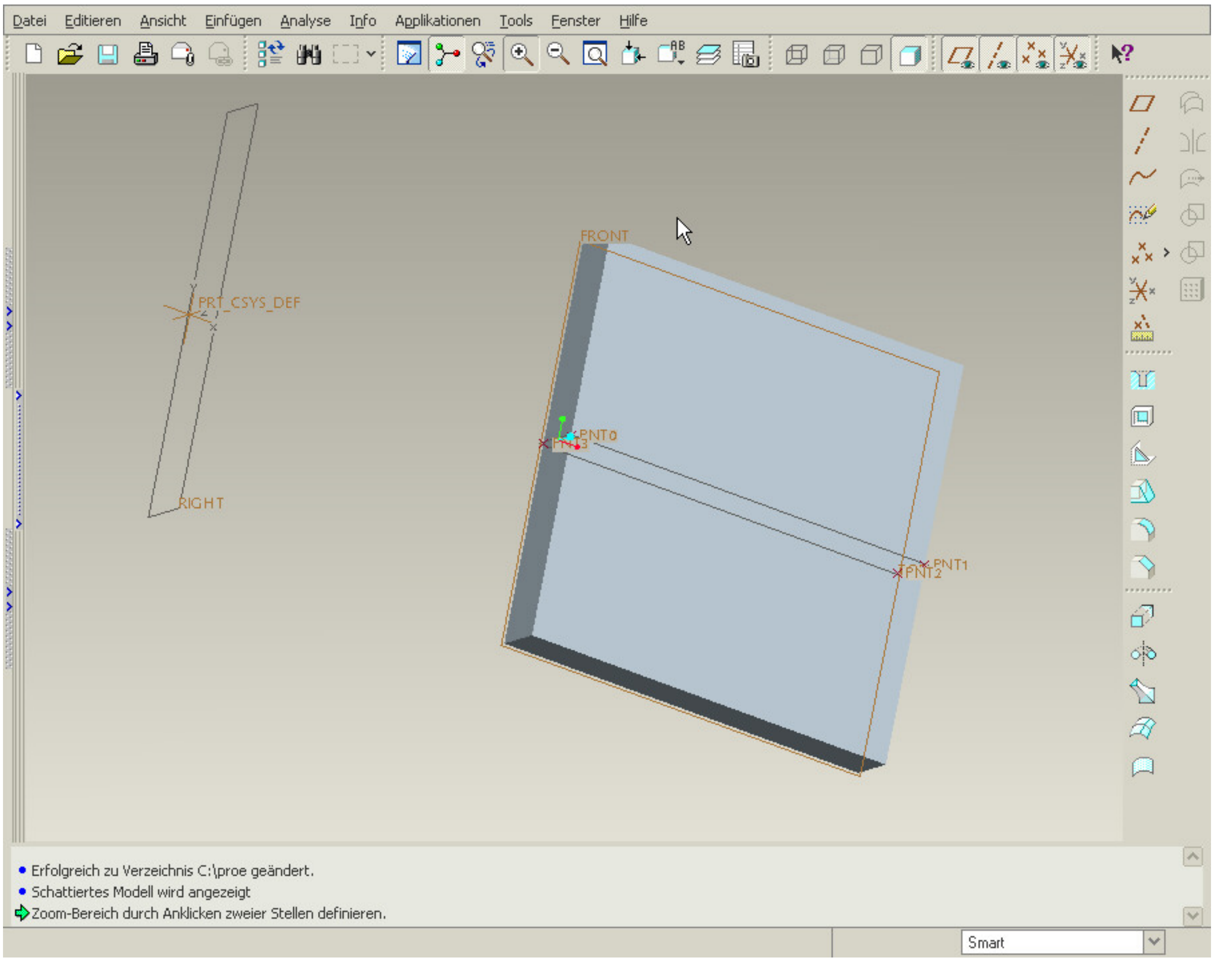
Choose file type before start. The UNIX version of Z88G operates in console mode.

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:



```
Erfolgreich zu Verzeichnis C:\proe geändert.
Schattiertes Modell wird angezeigt
Zoom-Bereich durch Anklicken zweier Stellen definieren.
```

(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 sparse matrix 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 3D 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) and – if needed – Z88I5.TXT (surface and pressure loads), files backups Z88I1.OLD, Z88I2.OLD and Z88I5.OLD (if needed) and computes the modified input files Z88I1.TXT and Z88I2.TXT and Z88I5.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 and –if needed – Z88I5.TXT. This can be done by:

- hand
- Z88 mesh generator Z88N (Z88I1.TXT only, then edit Z88I2.TXT and Z88I5.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 -T (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, Z88I2.TXT and Z88I5.TXT from the backup files Z88I1.OLD, Z88I2.OLD and Z88I5.OLD.

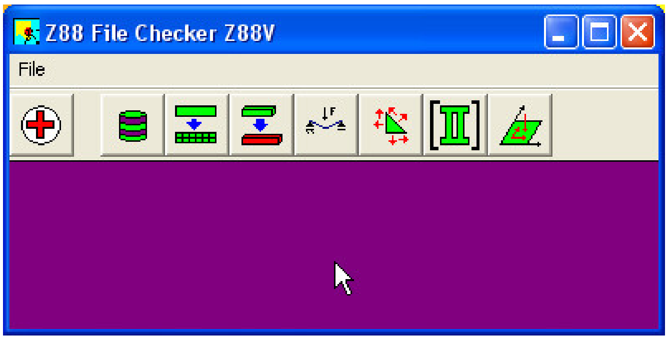
6) Enter the value of GS into Z88.DYN in the line MAXGS and lauch Z88F with a compute mode, e.g.

Windows: Z88F > Mode > Compute Mode, Compute > Go
UNIX: z88f -c (console) or Z88F -C (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 solvers Z88F and Z88I1/Z88I2), Z88I3.TXT (header file for the stress processor Z88D), Z88I4.TXT (header file for the iteration solver Z88I1/Z88I2), Z88I5.TXT (surface and pressure loads for the solvers) as well as the mesh generator input file Z88NI.TXT (for the mesh generator Z88N) for typing mistakes and logical faults. Cross-checks are executed, i. e. Z88I2.TXT, Z88I3.TXT and Z88I5.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 faulty 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 !

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 sparse matrix solvers Z88I1/Z88I2 and Z88I1/Z88PAR)*
- *Z88I5.TXT (surface and pressure loads –if needed)*
- *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. *Notepad* of Windows, *Emacs*, *Joe* at UNIX) or word processing program (e.g. *Word* at Windows). 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)?
Because you may 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 files Z88O5.TXT and Z88O8.TXT are not regular Z88 output files, containing the coordinates of the stress points and the reduced stresses for use for the plot program Z88O internally. It is a pure ASCII file, so that advanced users may use it for own routines, if necessary.

(3) Binary Files:
These files are used internally and are not for editing. They serve the fast data interchange between Z88 modules.

- *Z88O1.BNY*
- *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? No - 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 V11.0 and V12.0 are okay for Z88 V13.0, if the the plate flag and the surface and pressure loads flag are supplied.

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. If you won't see some pretty good stuff, then the rest can hardly be 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 may load without restriction 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 LINUX systems feature the converters *unix2dos* and *dos2unix*.

3.2 GENERAL STRUCTURE DATA Z88I1.TXT

Mind the following formats:

[Long]	= 4 bytes or 8 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)
Surface and pressure loads flag IQFLAG (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, 12 and 15 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 <u>1</u> 0 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 <u>1</u> 0*

Caution: This Z88 release allows only beams <u>or</u> plates in a structure, not both in the same structure, because the DOF of the beams and the plates are not compatible!

Explanation IQFLAG:
This flag controls if the surface and pressure loads file Z88I5.TXT is read (1) or not (0). The boundary conditions file Z88I2.TXT features constraints, defections and nodal forces. Surface and pressure loads may be defined in Z88I5.TXT, if needed.

- The following plane elements can deal with surface loads: plane stress elements No.7, 11 and 14, torus elements No.8, 12 and 15.
- The following plane elements can deal with surface and pressure loads: hexahedrons No.1 and 10 and tetrahedrons No.16 and 17.
- The plate elements No.18, 19 and 20 may read the pressure loads via Z88I5.TXT, (then set IPFLAG = 1 and IQFLAG = 1) but it is easier to define the pressure loads directly with the material entries of the file Z88I1.TXT (then set IPFLAG = 1 and IQFLAG = 0).

Example 1:
A threedimensional structure of tetrahedrons No.16 features 100 elements, 180 nodes, 540 DOF, 1 material information line, no change of coordinate system, no beams, no plates, use the surface and pressure loads file Z88I5.TXT.
>Thus: 3 180 100 540 1 0 0 0 <u>1</u>
Example 2:

A plate structure of elements No.18 features 1000 elements, 2000 nodes, 3000 DOF, 3 material information lines, no change of coordinate system, no beams, use the surface and pressure loads file Z88I5.TXT.
> Thus: 2 1000 2000 3000 3 0 0 1 1

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 and IQFLAG=0, 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 Z88O. 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 or 8 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 IBFLAG (must be 0 here !)
Plate flag IPFLAG (0 or 1)
Surface and pressure loads flag IQFLAG (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 IQFLAG:
You may set here IQFLAG=1 as a *reminder* if you plan to define a surface and pressure loads file Z88I5.TXT. However, IQFLAG has no effect for the mesh generator.

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 and IQFLAG=0, in addition:
surface 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:

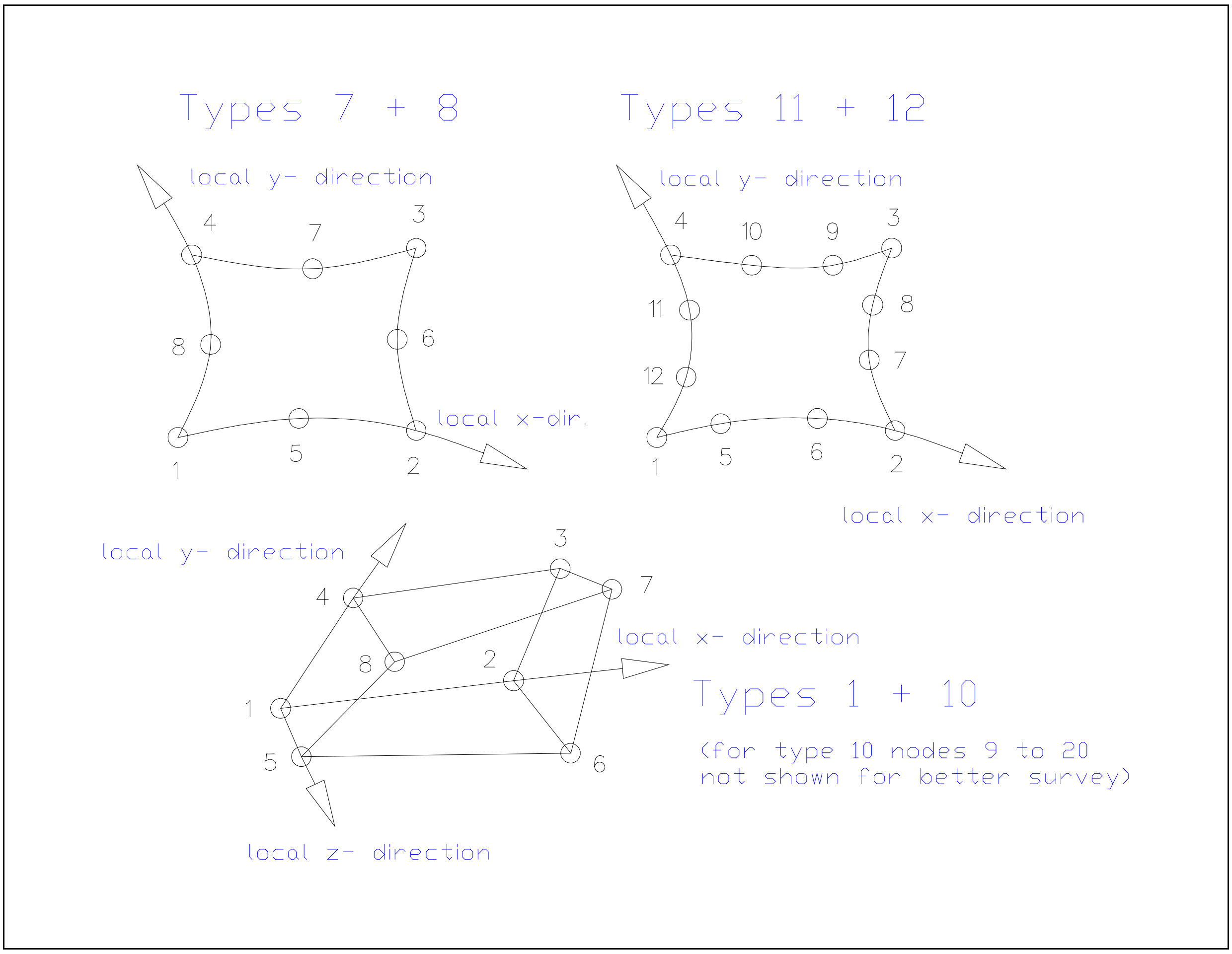
- "E": Subdivision equidistant, "e" is also permitted
- "L": Subdivision increasing geometrically in local coordinate direction
- "l": Subdivision decreasing geometrically in local coordinate direction

The local x-, y and z axes 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 below.

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 or 8 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 and surface loads pay attention to:

It is a good idea to define surface and pressure loads in the file Z88I5.TXT. However, for plates no.18, no.19 and no.20 you may define the surface load directly in the material information lines of Z88I1.TXT – which is much more convenient than via Z88I5.TXT!

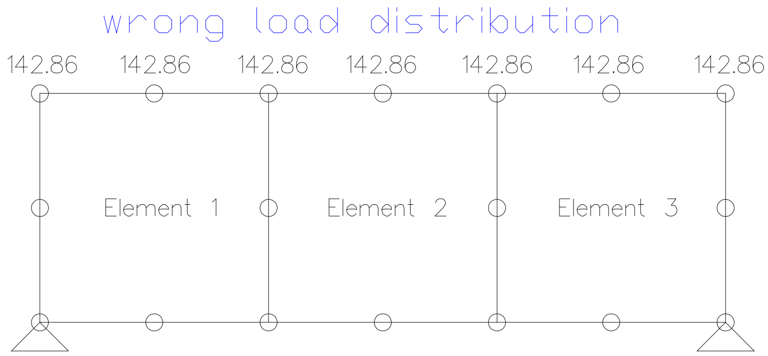
Only forces and constraints should entered here into Z88I2.TXT.

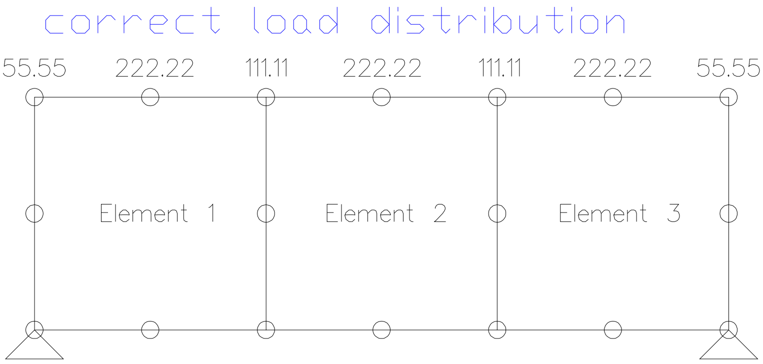
Of course, it is possible, too, to convert surface loads into single forces and to write these forces into Z88I2.TXT (which is the classical way but somewhat cumbersome).

For the elements with linear shape function, e.g. Hexahedrons No.1 and Torus No.6, edge 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:



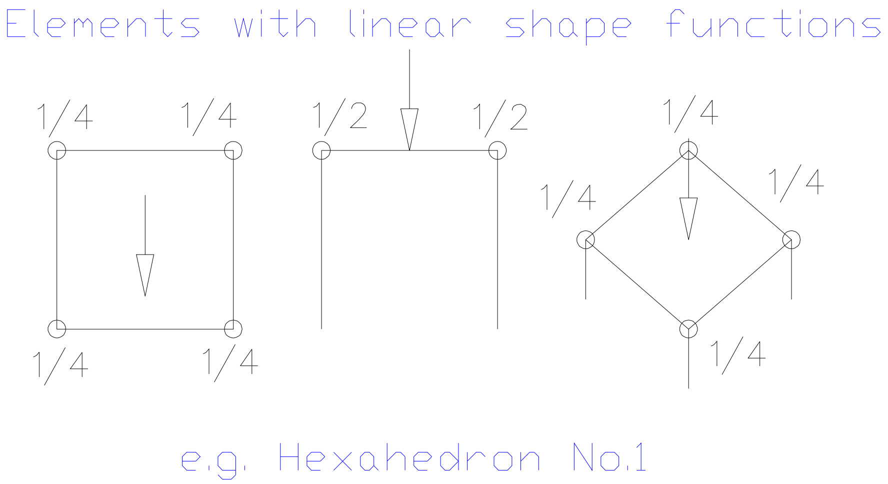


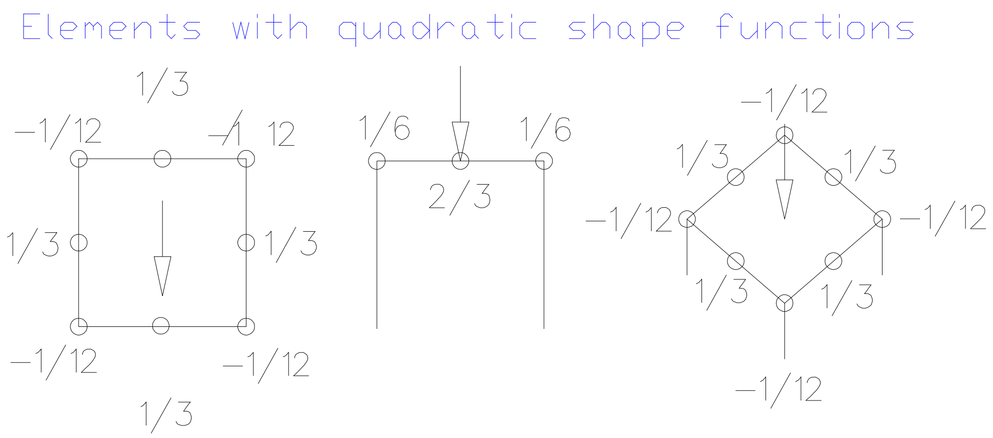
A FE structure consists of three plane stress elements No.7 with the load of 1,000 N distributed on the upper edge in Y direction. Above incorrect, below correct load sharing:

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

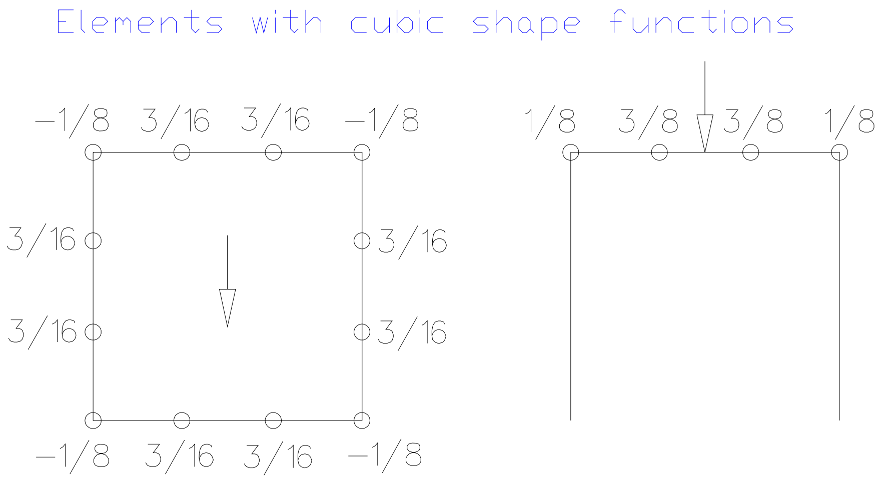
Correct: 2 × 1/6 + 2 × (1/6+1/6) + 3 × 2/3 = 18/6 = 3, corresponds to 1,000 N
"1/6 points" = 1,000/18×1 = 55.55
"2/6 points" = 1,000/18×2 = 111.11
"2/3 points" = 1,000/18×4 = 222.22

Control: 2×55.55 + 2×111.11 +3×222.22 = 1,000 N, o.k. Here's why:





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



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 or 8 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, reduced 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. Reduced 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. Reduced 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. Reduced 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
2 = principal or *Rankine* stresses
3 = *Tresca* 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 SPARSE MATRIX SOLVERS

Mind the following formats:
[Long] = 4 bytes or 8 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 limit 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 α [Double].*
This parameter for the SIC pre-conditioner of Z88I2 defines the Shift factor α (from 0 to 1, good values may vary from 0.0001 to 0.1. For further information consult the special literature)

4th entry: *Convergence acceleration parameter ω [Double].*
This parameter for the SOR pre-conditioner of Z88I2 defines the Relaxation factor ω (from 0 to 2, good values may vary from 0.8 to 1.2. How to choose ω ? Good question! Try 1.0 for a first start and try other values for further Z88 runs with this structure.

5th entry: Z88PAR: number of cores or CPUs on multi-core computers. A maximum number of 9 is allowed.

Example 1: You want to stop after 5000 iterations, you choose a limit of 0.0000001 and the convergence acceleration parameter ω will be 0.9 for use with SORCG solver.
> Thus: 5000 0.0000001 0.001 0.9 1

Example 2: You want to use the Sparse Matrix Iteration Solver Z88I2. You want to stop positively after 10000 iterations, the limit shall be 10^{-9} and the Shift factor α for SIC shall be 0.001 because you want to use the SIC pre-conditioner.
> Thus: 10000 1e-9 0.001 0.9 1

Example 3: *You want to use the direct Sparse Matrix Solver with fill-in Z88PAR and you have two double core CPUs in your computer installed.*
> Thus: 10000 1e-9 0.001 0.9 4
The not-underlined entries have no meaning for Z88PAR.

3.7 SURFACE/PRESSURE LOADS FILE Z88I5.TXT

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

1st Input group, i.e. the first line contains
number of surface and pressure loads [Long]

2nd Input group, i.e. the second and following lines contain

surface and pressure loads – one line per load. Of course, an element may have more than one load applied.

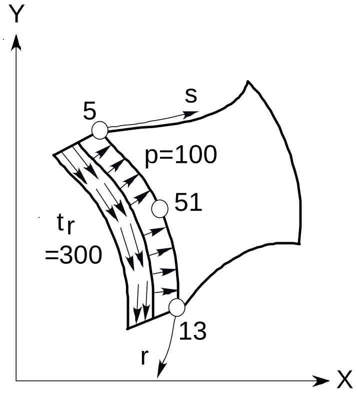
The following entries depend from the element type with surface and pressure load to avoid unnessesary data entries. Define the the local r and s directions by the nodes and their sequence. These local directions for the surface loads may differ from the local r and s coordinate system of the finite element. The numbering has to conform to the element numbering, see chp. 4. Separate each item by at least one blank.

→ Plain stress element No.7 and 14 and Torus elements No.8 and 15:

Element number with surface load [Long]
Pressure, positive if pointing towards the edge [Double]
Tangential shear, positive in local r-direction [Double]
3 nodes of the loaded edge [3 × Double]

Example: The plain stress element 97 features surface load. The load should be applied onto the edge defined by the corner nodes 5 and 13 and by the mid node 51. One surface load is applied normally to the edge with 100 N/mm and the other surface load is applied tangentially and positive in local r direction with 300 N/mm (defined by the two corner nodes). Thus:

> 97 100. 300. 5 13 51



→ Hexahedron No.1:

Element number with surface and pressure load [Long]
Pressure, positive if pointing towards the surface [Double]
Tangential shear, positive in local r direction [Double]
Tangential shear, positive in local s direction [Double]
4 nodes of the loaded surface [4 × Long]

Example: The hexahedron 356 features surface loads. The load should be applied onto the surface defined by the corner nodes 51, 34, 99 and 12 .The first surface load is pressure with 100 N/mm. The second surface load is applied tangentially and positive in local r direction with 200 N/mm. The third surface load is applied tangentially and positive in local s direction with 300 N/mm . Thus

> 356 100. 200. 300. 51 34 99 12

→ Hexahedron No.10:

Element number with surface and pressure load [Long]
Pressure, positive if pointing towards the surface [Double]
Tangential shear, positive in local r direction [Double]
Tangential shear, positive in local s direction [Double]
8 nodes of the loaded surface [8 × Double]

Example: The hexahedron 456 features surface loads. The load should be applied onto the surface defined by the corner nodes 51, 34, 99 and 12 and the mid nodes 102, 151, 166 and

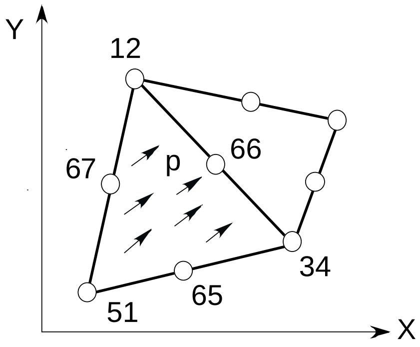
191 .The first surface load is pressure with 100 N/mm. The second surface load is applied tangentially and positive in local r direction with 200 N/mm. The third surface load is applied tangentially and positive in local s direction with 300 N/mm . Thus
> 456 100. 200. 300. 51 34 99 12 102 151 166 191

→ **Tetrahedron No.17:**
Element number with pressure load [Long]
Pressure, positive if pointing towards the surface [Double]
3 nodes of the loaded surface [3 × Double]

Example: The tetrahedron 356 features surface loads. The load should be applied onto the surface defined by the corner nodes 51, 34 and 12.The surface load is pressure with 100 N/mm pointing towards the surface, i.e. positive. Thus:
> 356 100. 51 34 12

→ **Tetrahedron No.16:**
Element number with pressure load [Long]
Pressure, positive if pointing towards the surface [Double]
6 nodes of the loaded surface [6 × Double]

Example: The tetrahedron 888 features surface loads. The load should be applied onto the surface defined by the corner nodes 51, 34 and 12 and the mid nodes 65, 66 and 67.The surface load is pressure with 100 N/mm pointing towards the surface, i.e. positive. Thus:
> 888 100. 51 34 12 65 66 67



→ **Plate elements No.18, 19 and 20:**
Element number with pressure load [Long]
Pressure, positive if pointing towards the surface [Double]
It is easier to enter the pressure loads for plate elements directly into Z88I1.TXT than via Z88I5.TXT

4 DESCRIPTION OF THE FINITE ELEMENTS

4.1 HEXAHEDRON NO.1 WITH 8 NODES

4.2 BEAM NO.2 WITH 2 NODES IN SPACE

4.3 PLANE STRESS TRIANGLE NO.3 WITH 6 NODES

4.4 TRUSS NO.4 WITH 2 NODES IN SPACE

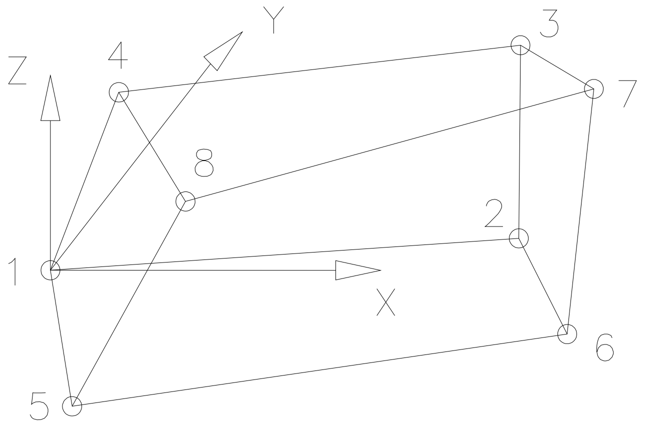
4.5 CAM NO.5 WITH 2 NODES

4.6 TORUS NO.6 WITH 3 NODES

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 axes 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 mesh 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)
> IQFLAG=1 if surface and pressure loads for this element are filed in Z88I5.TXT
> 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.

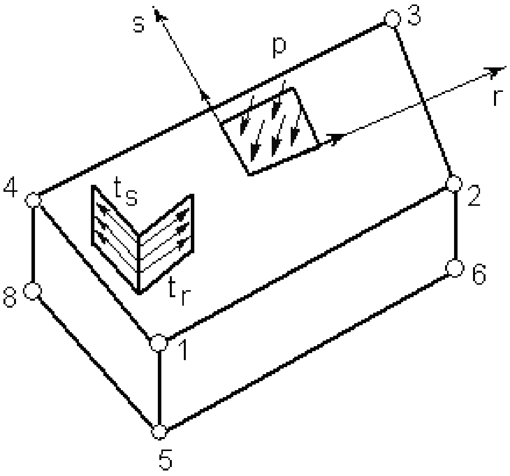
Z88I3.TXT
> Integration order INTORD for stress calculation:
Can be different from INTORD in Z88I1.TXT.
0 = Calculation of stresses in the corner nodes
1,2,3,4 = Calculation of stresses in the Gauss points
> any *KFLAG,* has no influence

> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses in the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses in the Gauss points (INTORD not 0 !)
3 = Tresca stresses in the Gauss points (INTORD not 0 !)

Z88I5.TXT
This file is optional and only used if in addition to nodal forces surface and pressure loads applied onto element no.1:

> Element number with surface and pressure load
> Pressure, positive if pointing towards the surface
> Tangential shear, positive in local r direction
> Tangential shear, positive in local s direction
> 4 nodes of the loaded surface

The local r direction is defined by the nodes 1-2, the local s direction is defined by the nodes 1-4. The local nodes 1, 2, 3 , 4 may differ from the local nodes 1, 2, 3, 4 used for the coincidence.

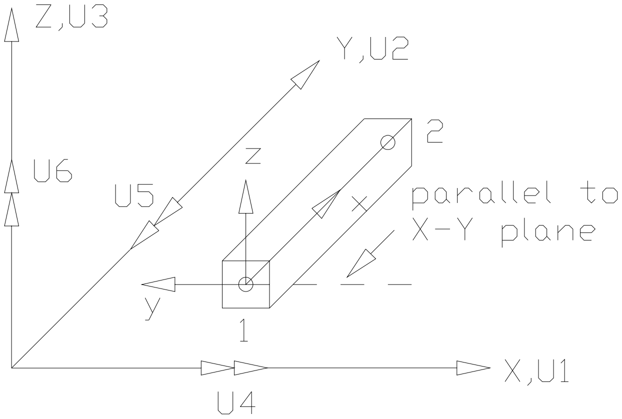


Results:

Displacements in X, Y and Z
Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises or principal or Tresca 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*

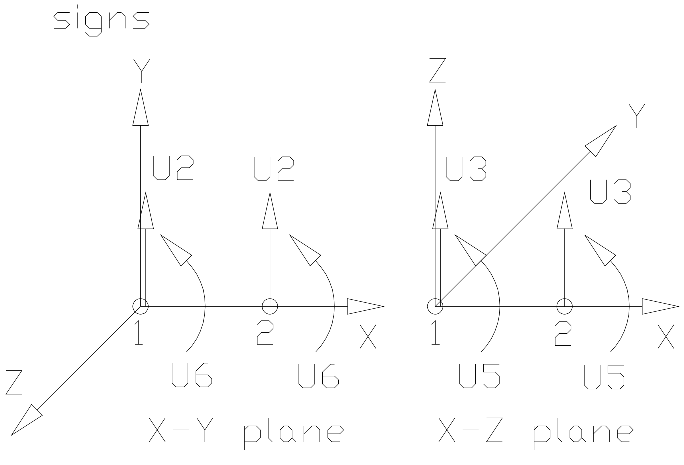
Z88I1.TXT

> 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

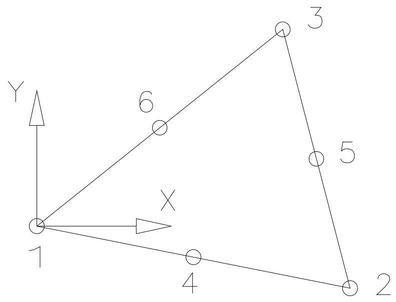
Z88I3.TXT
Beams No.2 have no influence. However, Z88I3.TXT must exist (with any content).

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. This element is obsolete and kept in Z88 only for studies. Elements No. 7, 11 or 14 are much more better. Pay attention to edge loads, cf. chapter 3.4. No entries into the surface and pressure loads file Z88I5.TXT!



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

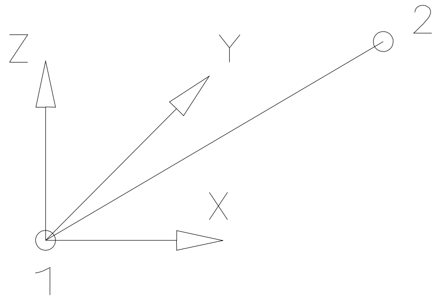
> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses in the centre of gravity
2 = principal or Rankine stresses in the centre of gravity
3 = Tresca stresses in the centre of gravity

Results:

Displacements into X and Y
Stresses: The stresses are calculated in the element's centre of gravity. The coordinates of the centre 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 centre of gravity are printed. Optional von Mises stresses in the centre 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*

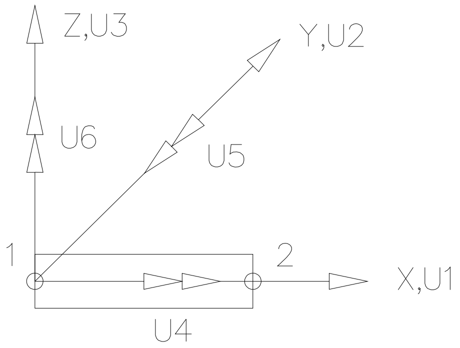
Z88I1.TXT
> KFLAG for cartesian (0) or cylindrical coordinates (1)
> 3 degrees of freedom for each node
> Element type is 4
> 2 nodes per element
> Cross-section parameter QPARA is the cross-sectional area of the truss

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:

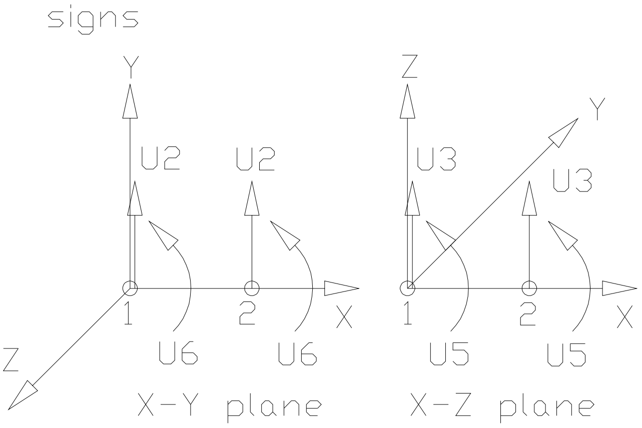
CAD (see chapter 2.7.2): *Line from node 1 to node 2*

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:

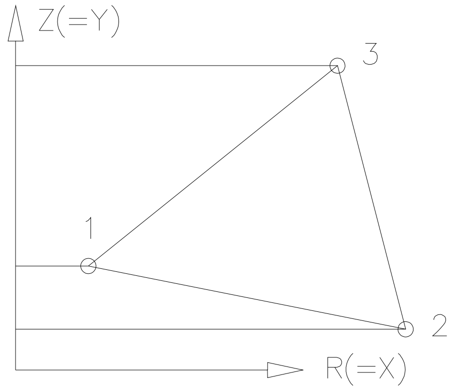
Deflections in X, Y and Z and rotations around X, Y and Z. Attention DOF5 (not right hand rule), see below
Stresses: SIGXX, TAUXX: Direct stress, shear stress, SIGXY1, SIGXY2: Bending stress in X-Y plane for node 1 and node 2, SIGXZ1 SIGXZ2: Bending stress in X-Z plane for node 1 and node 2
Nodal forces 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.12 or No.15. No entries into the surface and pressure loads file Z88I5.TXT!

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 centre of gravity. However, the use of the torus elements No.8 or No.12 or No.15 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 centre of gravity
2 = principal or Rankine stresses plotted in the centre of gravity
3 = Tresca stresses plotted in the centre of gravity

Results:

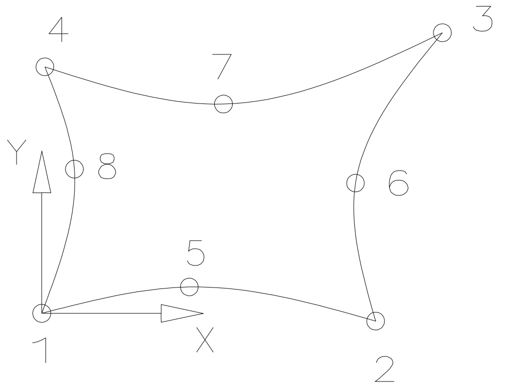
Displacements in R and Z (= X and Y)
Stresses: The stress are internally computed in the corner nodes, but plotted in the centre of gravity.
It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), 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 axes 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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT. You may combine this element with elements no.3 (not recommended) or elements no.14 (good).
Plane Stress Elements No.7 can be generated by the mesh 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)
> IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT
> 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

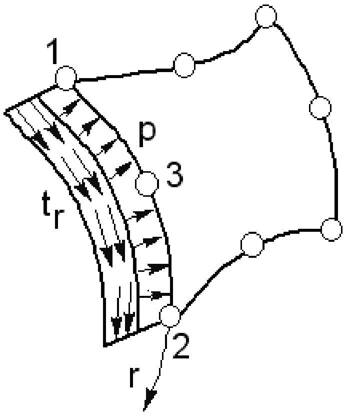
> *KFLAG* = 0: Calculation of SIGXX, SIGYY and TAUXY
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT
This file is optional and only used if in addition to nodal forces edge loads applied onto element no.7:

> Element number with surface and pressure load
> Pressure, positive if pointing towards the edge
> Tangential shear, positive in local r direction
> 2 corner nodes and one mid node of the loaded surface

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3 may differ from the local nodes 1, 2, 3 used for the coincidence.



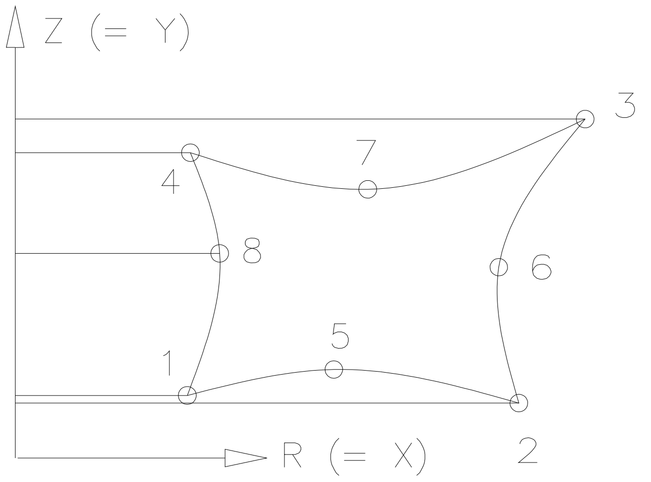
Results:

Displacements in 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 or pricipal or Tresca 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 axes 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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT. You may combine this element with elements no.15.

Torus elements No.8 can be generated by the mesh 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
> IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT
> 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

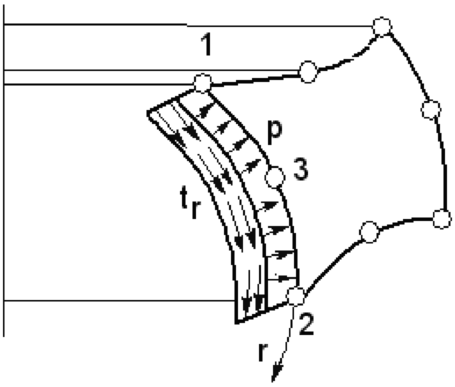
> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and only used if in addition to nodal forces edge loads applied onto element no.8:

> Element number with surface and pressure load
> Pressure, positive if pointing towards the edge
> Tangential shear, positive in local r direction
> 2 corner nodes and one mid node of the loaded surface

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3 may differ from the local nodes 1, 2, 3 used for the coincidence.

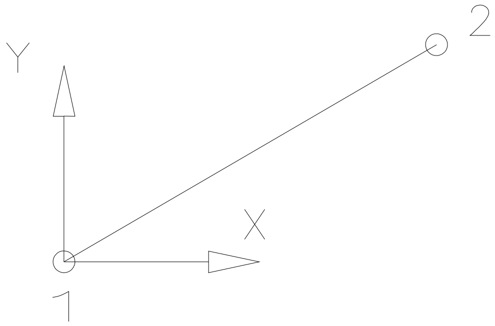


Results:

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:

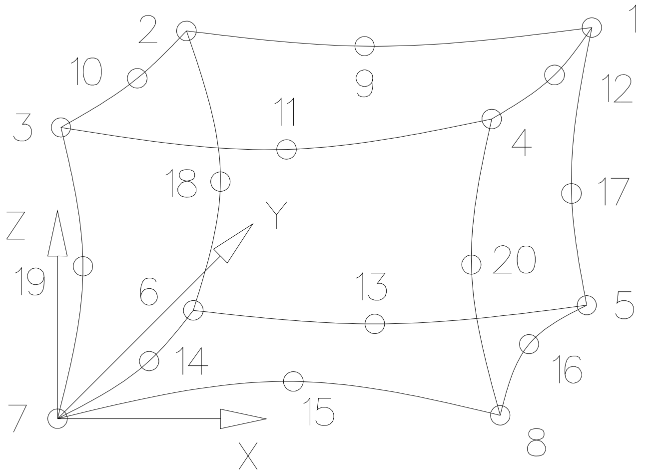
Displacements in X and Y
Stresses: Normal stresses
Nodal forces in X and Y for each element and each node.

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 axes 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. Pay attention to surface and pressure loads when using forces, cf. chapter 3.4. It is easier to enter these loads via the surface and pressure loads file Z88I5.TXT.



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 mesh 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.

<u>Hexahedron No.10 is recommended for all sort of deflection and stress computation in space</u>. Though its need for memory and computing power is enormous, this element gives precise results for displacements and stresses. Or use it as super elements 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)
> IQFLAG=1 if surface and pressure loads for this element are filed in Z88I5.TXT
> 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

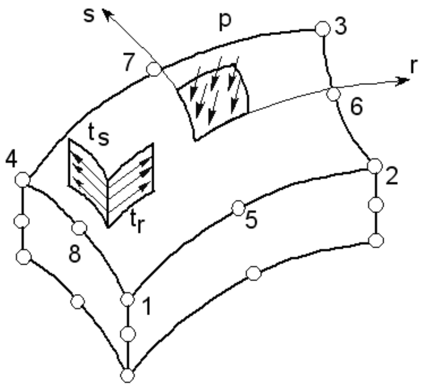
> KFLAG , any, has no influence

> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT
This file is optional and only used if in addition to nodal forces surface and pressure loads applied onto element no.10:

> Element number with surface and pressure load
> Pressure, positive if pointing towards the surface
> Tangential shear, positive in local r direction
> Tangential shear, positive in local s direction
> 4 nodes of the loaded surface

The local r direction is defined by the nodes 1-2, the local s direction is defined by the nodes 1-4. The local nodes 1, 2, 3 , 4 may differ from the local nodes 1, 2, 3, 4 used for the coincidence.



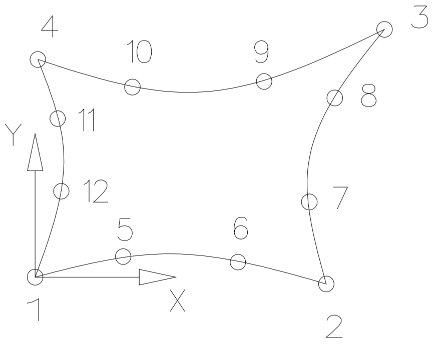
Results:

Displacements in X, Y and Z
Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises or principal or Tresca 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 axes 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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT.

Plane Stress Elements No.7 can be generated by the mesh 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 mesh 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)
> IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT
> 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

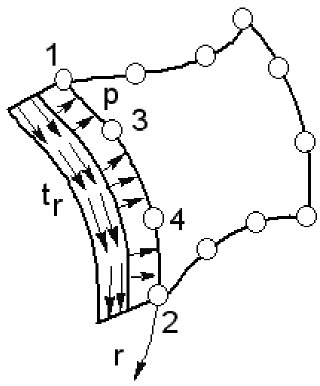
> KFLAG = 0: Calculation of SIGXX, SIGYY and TAUXY
> KFLAG = 1: Additional calculation of SIGRR, SIGTT and TAURT

> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT
This file is optional and only used if in addition to nodal forces edge loads applied onto element no.11:

> Element number with surface and pressure load
> Pressure, positive if pointing towards the edge
> Tangential shear, positive in local r direction
> 2 corner nodes and 2 mid nodes of the loaded surface

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3, 4 may differ from the local nodes 1, 2, 3, 4 used for the coincidence.



Results:

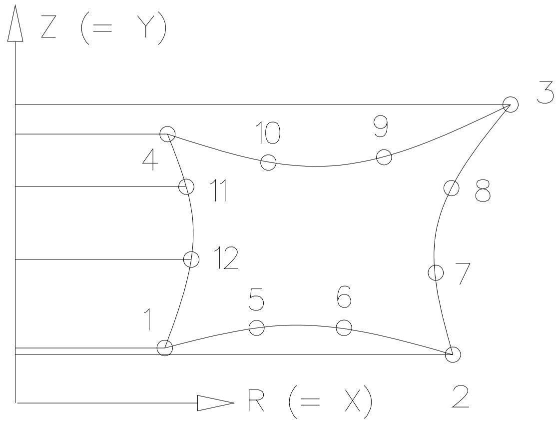
Displacements in X and Y.
Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. For KFLAG = 1 the radial stresses SIGRR, the tangential stresses SIGTT

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 or principal or Tresca 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 axes 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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT.

Torus elements No.8 can be generated by the mesh 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 mesh 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
> IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT
> 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

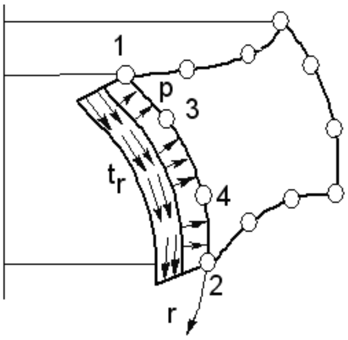
> *KFLAG* , any, has no influence

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

Z88I5.TXT
This file is optional and only used if in addition to nodal forces edge loads applied onto element no.12:

> *Element number with surface and pressure load*
> *Pressure, positive if pointing towards the edge*
> *Tangential shear, positive in local r direction*
> *2 corner nodes and 2 mid nodes of the loaded surface*

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3, 4 may differ from the local nodes 1, 2, 3, 4 used for the coincidence.



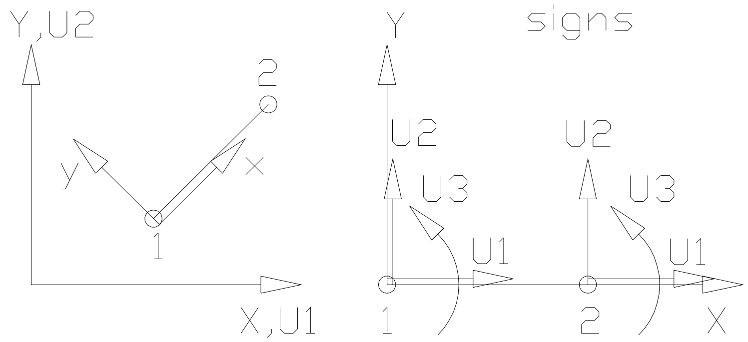
Results:

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 or principal or Tresca 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).

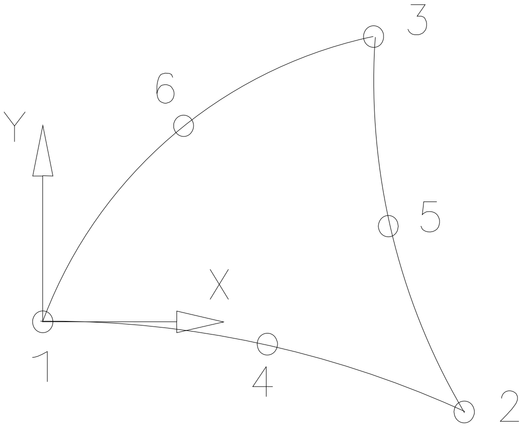
Results:

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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT.

This element type is implemented for use with automeshers e.g. Pro/MECHANICA for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesh 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)*
> *IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT*
> *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 Gauss points. 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

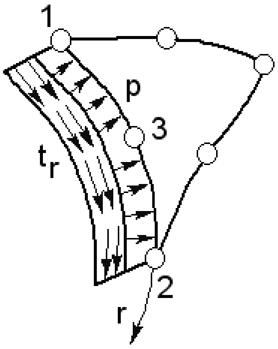
> *KFLAG* = 1: Additional calculation of SIGRR, SIGTT and TAURT

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

Z88I5.TXT
This file is optional and only used if in addition to nodal forces edge loads applied onto element no.14:

> *Element number with surface and pressure load*
> *Pressure, positive if pointing towards the edge*
> *Tangential shear, positive in local r direction*
> *2 corner nodes and one mid node of the loaded surface*

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3 may differ from the local nodes 1, 2, 3 used for the coincidence.



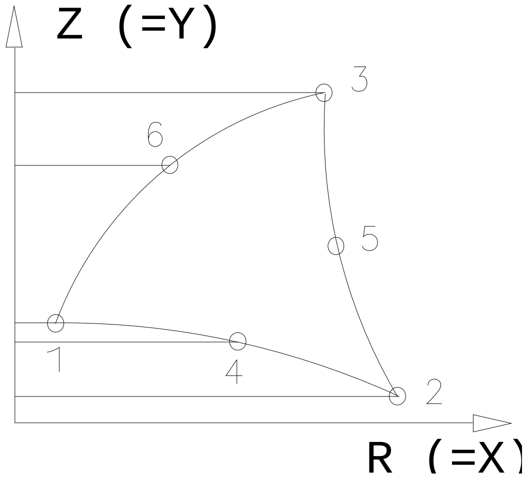
Results:

Displacements in 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 or principal or Tresca 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 when using forces, cf. chapter 3.4. It is easier to enter edge loads via the surface and pressure loads file Z88I5.TXT.

This element type is implemented for use with automeshers e.g. Pro/MECHANICA for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesh 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.



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
> IQFLAG=1 if edge loads for this element are filed in Z88I5.TXT
> 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 Gauss points. 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:*

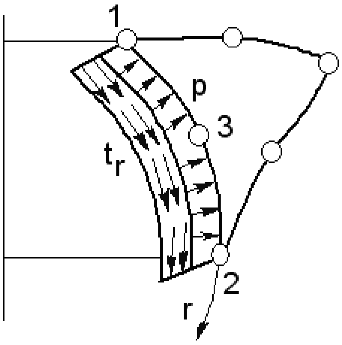
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and only used if in addition to nodal forces edge loads applied onto element no.15:

> *Element number with surface and pressure load*
> *Pressure, positive if pointing towards the edge*
> *Tangential shear, positive in local r direction*
> *2 corner nodes and one mid node of the loaded surface*

The local r direction is defined by the nodes 1-2. The local nodes 1, 2, 3 may differ from the local nodes 1, 2, 3 used for the coincidence.



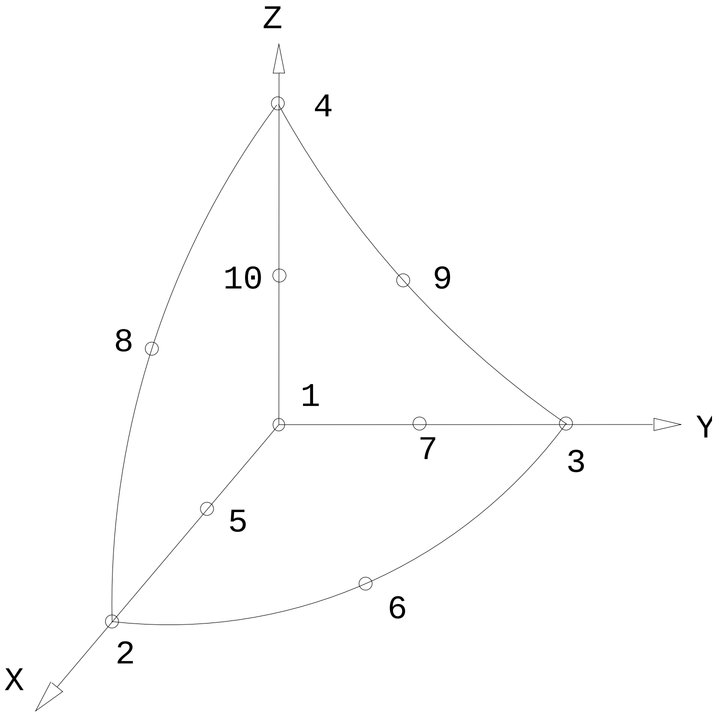
Results:
Displacements in R and Z (= X and Y).
Stresses: The stresses are calculated in the corner nodes or Gauss points and printed along with their locations. It is: SIGRR = stress in R direction = radial stress (= X direction), SIGZZ = stress in Z direction (= Y direction), TAURZ = shear stress in RZ plane (= XY plane), SIGTE = stress in peripherical direction = tangential stress. Optional von Mises or principal or Tresca 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. Pay attention to pressure loads when using forces, cf. chapter 3.4. It is easier to enter pressure loads via the surface and pressure loads file Z88I5.TXT.

This element type is implemented for use with automeshers e.g. Pro/MECHANICA for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesh 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 mesh 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)
> IQFLAG=1 if pressure loads for this element are filed in Z88I5.TXT
> 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

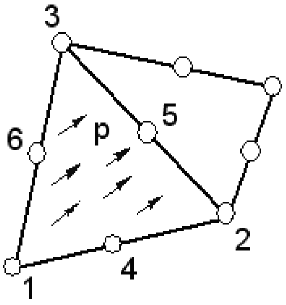
> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and only used if in addition to nodal forces pressure loads applied onto element no.16:

> Element number with pressure load
> Pressure, positive if pointing towards the edge
> 3 corner nodes and 3 mid nodes of the loaded surface

The local nodes 1 to 6 may differ from the local nodes 1 to 6 used for the coincidence.



Results:
Displacements in X, Y and Z
Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises or principal or Tresca stresses.
Nodal forces in X, Y and Z for each element and each node.

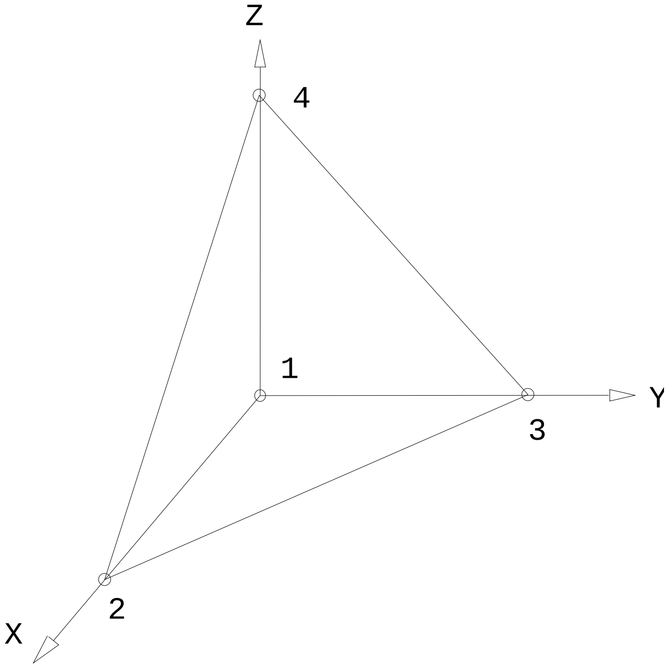
4.17 TETRAHEDRON NO.17 WITH 4 NODES

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/MECHANICA for the 3D CAD system Pro/ENGINEER by Parametric Technology. Thus, a mesh 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 mesh 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)
> IQFLAG=1 if pressure loads for this element are filed in Z88I5.TXT
> 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)

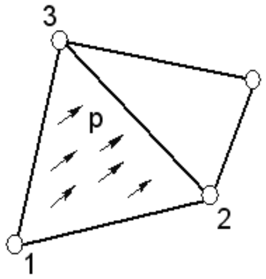
> KFLAG , any, has no influence

> Reduced stress flag ISFLAG:
0 = no calculation of reduced stresses
1 = von Mises stresses computed for the Gauss points (INTORD not 0 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT
This file is optional and only used if in addition to nodal forces pressure loads applied onto element no.17:

> Element number with pressure load
> Pressure, positive if pointing towards the edge
> 3 corner nodes of the loaded surface

The local nodes 1 to 3 may differ from the local nodes 1 to 3 used for the coincidence.



Results:

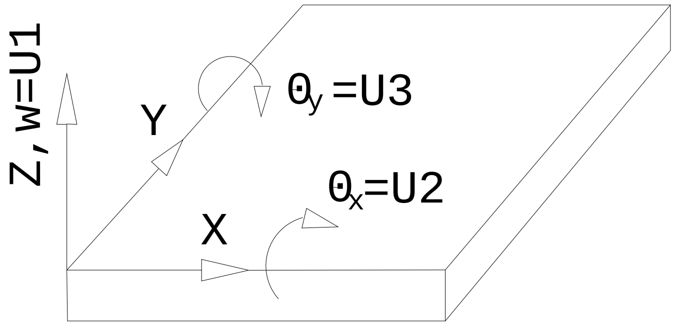
Displacements in X, Y and Z
Stresses: SIGXX, SIGYY, SIGZZ, TAUXY, TAUYZ, TAUZX, respectively for corner nodes or Gauss points. Optional von Mises or principal or Tresca stresses.
Nodal forces in X, Y and Z for each element and each node.

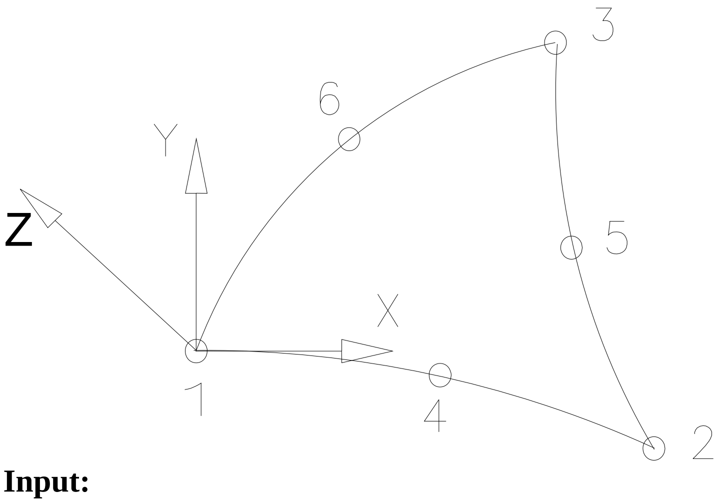
4.18 PLATE NO.18 WITH 6 NODES

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 axes 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/MECHANICA 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)
> set surface and pressure loads flag IQFLAG to 0 for your convenience. Then the entry of the pressure is done via the "Second moment of inertia RIYY", see below. If IQFLAG is set to 1, then the entry of the pressure is done via the surface and pressure loads file Z88I5.TXT
> 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 pressure 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 Gauss points. 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 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and normally not used here because it is much more convenient to enter the pressure data for the plate elements into Z88I1.TXT in the section material information. However, the possibility for entering the pressure loads by the surface and pressure loads file Z88I5.TXT, too, is implemented for universal use of this file. Then set IQFLAG to 1 and proceed as follows:

> Element number with pressure load
> Pressure, positive if pointing towards the edge

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 × length / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force × 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* or principal or Tresca stresses.
Nodal forces in X and Y for each element and each node.

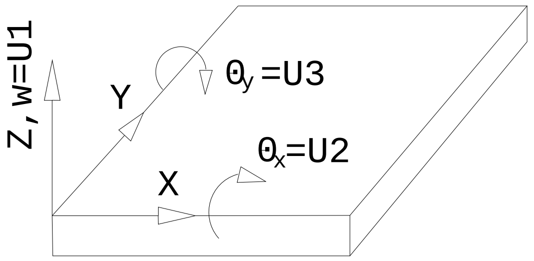
4.19 PLATE NO.19 WITH 16 NODES

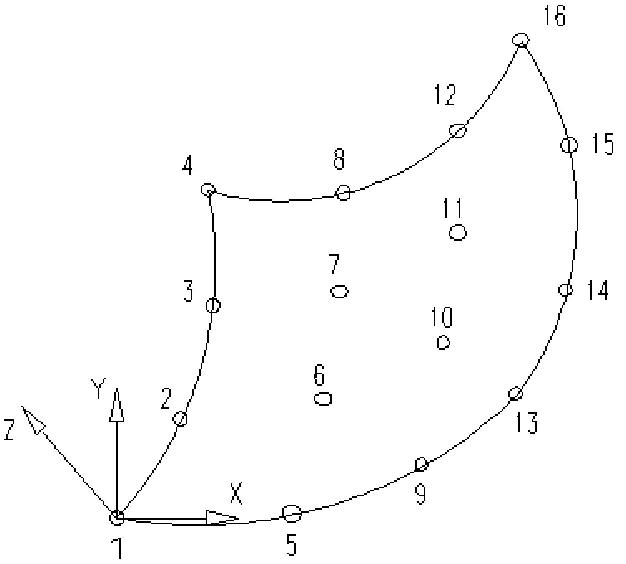
This is a curvilinear Lagrange-*Reissner-Mindlin* plate element with cubic shape functions. The transformation is isoparametric. The integration is carried out numerically in both axes according to Gauss-Legendre. Consequently, the integration order can be selected in Z88I1.TXT in the material information lines. The order 4 (= 4 × 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 quite fine.
For example, some lines from a mesh generator input file Z88NI.TXT:

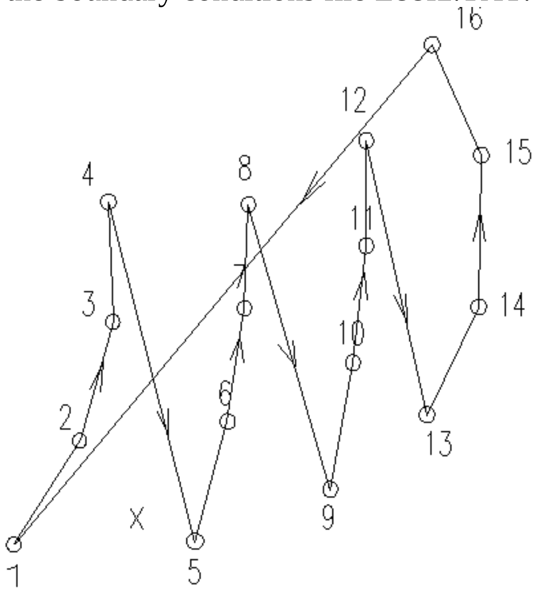
```
 .....
5    20          super element 5 of type 20
20   25   27   22   24   26   28   21
.....
5    19          generate from super element 5 (which is of type 20 is, see above) finite elements of type 19
3E   3E      .. and subdivide them three times equidistant in X-direction and three times equidistant 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)

> set surface and pressure loads flag IQFLAG to 0 for your convenience. Then the entry of the pressure is done via the "Second moment of inertia RIYY", see below. If IQFLAG is set to 1, then the entry of the pressure is done via the surface and pressure loads file Z88I5.TXT

> 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 surface 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 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and normally not used here because it is much more convenient to enter the pressure data for the plate elements into Z88I1.TXT in the section material information. However, the possibility for entering the pressure loads by the surface and pressure loads file Z88I5.TXT, too, is implemented for universal use of this file. Then set IQFLAG to 1 and proceed as follows:

> *Element number with pressure load*
> *Pressure, positive if pointing towards the edge*

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:

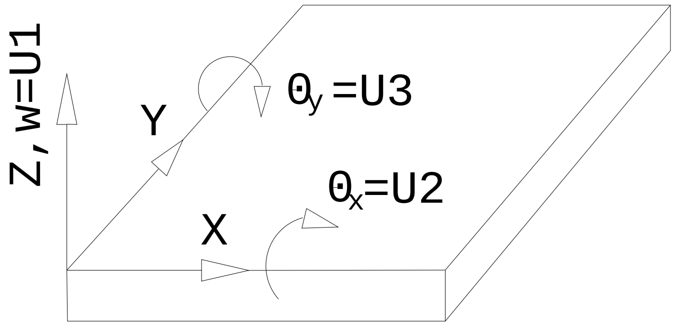
- plate bending moments M_{xx} and M_{yy} (unit: force × length / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force × length / length)
- the shear forces Q_{yz} and Q_{zx} (unit: force / length)
- the true stresses resulting from plate bending moments and plate torsion moments

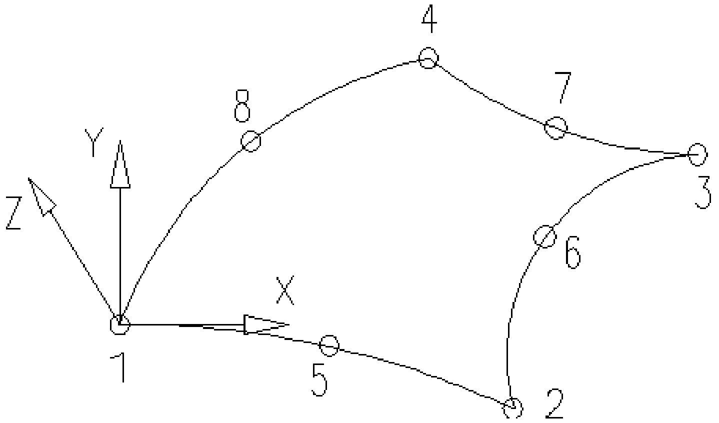
Optional *von Mises* or principal or Tresca stresses.
Nodal forces in X and Y for each element and each node.

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 axes 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/MECHANICA 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)*
> *set surface and pressure loads flag IQFLAG to 0 for your convenience. Then the entry of the pressure is done via the "Second moment of inertia RIYY", see below. If IQFLAG is set to 1, then the entry of the pressure is done via the surface and pressure loads file Z88I5.TXT*
> *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 !)
2 = principal or Rankine stresses computed for the Gauss points (INTORD not 0 !)
3 = Tresca stresses computed for the Gauss points (INTORD not 0 !)

Z88I5.TXT

This file is optional and normally not used here because it is much more convenient to enter the pressure data for the plate elements into Z88I1.TXT in the section material information. However, the possibility for entering the pressure loads by the surface and pressure loads file Z88I5.TXT, too, is implemented for universal use of this file. Then set IQFLAG to 1 and proceed as follows:

> Element number with pressure load
> Pressure, positive if pointing towards the edge

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 × length / length)
- plate torsion moments $M_{xy} = M_{yx}$ (unit: force × 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* or principal or Tresca 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 CD or the Internet distribution. 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. Vary the input files, especially the mesh generator-input files for the examples 1, 5 and 7. Doing so gives you a smart feeling for the how-to of Z88 very quick.

If examples won't 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 many examples do run even on old-fashioned computers. Nevertheless, Z88 is running very large structures on modern PCs without any problems, see example 5.10. The largest structure computed with Z88 up to now featured 4.5 Mio. DOF and was run on a 64 Bit PC with 64 Bit Windows Server 2003 and with 64 Bit LINUX, too. 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 2.7.2. Did you "snap" the points cleanly ? If nothing works try another CAD program.

If you've got a 3D CAD program with an integrated automesher you may export FE meshes to COSMOS or NASTRAN files and read these files into Z88 with Z88G. Check the amount of needed memory and the quality of the nodal numbering by running Z88F in test mode. Renumber with the Cuthill McKee program Z88H, if necessary. Or, even better, use one of the sparse matrix solvers.

Example 1: Fork wrench. Plane stress problem with Serendipity Plane Stress No.7 and mesh generator use. Learning objectives: CAD and mesh generator use at curvilinear plane structures, displaying stresses in the plot program. This example is fixed on the Z88 distribution 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 over-defined. 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 over-defined. 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 mesh generation of Hexahedrons No.1 (8 nodes). Learning objectives: Use of the mesh 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 mesh 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 and **surface loads**, showing stresses in the plot program.

Example 7: Press fit. Axially symmetric problem with Tori No.8 and use of mesh generator. Learning objectives: Work with torus elements, use of the mesh generator with mesh 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 3D converter Z88G, the Cuthill-McKee program Z88H and both the solvers, i.e. the direct Cholesky solver Z88F and the sparse matrix 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 sparse matrix iteration solver.

Example 10: Piston of a diesel engine with Tetrahedrons No.16. Learning objectives: Starting with a NASTRAN file from Pro/ENGINEER, we will use the 3D converter Z88G and the sparse matrix iteration solver Z88I1/Z88I2. This is an example for a FEA structure imported from a CAD system using the **surface and pressure loads file Z88I5.TXT**.

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 formatting 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 de-facto 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 CDs or Internet packages 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: Press button *Z88X* in the Z88 commander, Button *DXF → Z88NI, Run Button.* Close program Z88X.

LINUX/UNIX: In the Z88 commander under *CAD converters* press button *DXF → Z88NI.*

Z88O, looking at the super structure.
Windows: In the Z88 commander press button *Z88O.* If you press now the *Run button*, don't worry about the error message because Z88O wants to load the file Z88I1.TXT as a default which does not exist at the moment. However, you want to load now Z88NI.TXT: Proceed as follows: *Diskette button* > *Z88NI.TXT*, *Run button.* Then switch to *Wireframe* by the appropriate button and show the nodal numbers and the element numbers by *Menu* > *Labels* > *All*. Zoom in with the *Prior* key. Close Z88O.

LINUX/UNIX: In the Z88 commander press button *Z88O.* If you press now the *Run button*, don't worry about the error message because Z88O wants to load the file Z88I1.TXT as a default which does not exist at the moment. However, you want to load now Z88NI.TXT: Proceed as follows: *File button* > *Z88NI.TXT*, *Run button.* Then switch to *Wireframe* by the appropriate button and show the nodal numbers and the element numbers by *Menu* > *Labels* > *All*. Zoom in with the *Prior* key. Close Z88O.

Z88N, mesh generator, reads Z88NI.TXT and produces Z88I1.TXT.
Windows: In the Z88 commander button *Z88N*, *Run button.* Close Z88N. **Hint:** You should always close the unneeded Z88 modules to have a maximum of memory.

LINUX/UNIX: In the Z88 commander press button *Z88N.*

Z88O, looking at the finite elements structure. Proceed as follows:

Windows: In the Z88 commander press button *Z88O, Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88O*, *Run Button.*

Z88F, calculates displacements. Proceed as follows:
Windows: In the Z88 commander press button *Z88F, CD Button* (is default), *Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88F -C.*

Z88D, calculates stresses. Proceed as follows:
Windows: In the Z88 commander press button *Z88D, Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88D.*

Z88O, looking at the deflected finite elements structure. Proceed as follows:

Windows: In the Z88 commander press button *Z88O*, *Run Button*, *Wireframe Button*, *Deflected Button.* As a default, the deflections are multiplied by 100. This is too large for this example. Thus: *Menu* > *Factors* > *Deflections* > enter for FUX and FUY *10* . You may also look at the reduced stresses because Z88D was run before. Try the buttons *Reduced stresses in corner nodes*, *Reduced stresses mean values per element* and *Reduced stresses in Gauss points* (this feature for undeflected structures only).

LINUX/UNIX: In the Z88 commander press button *Z88O*, *Run Button*, *Wireframe Button*, *Deflected Button.* As a default, the deflections are multiplied by 100. This is too large for this example. Thus: *Menu* > *Factors* > *Deflections* > enter for FUX and FUY *10* . You may also look at the reduced stresses because Z88D was run before. Try the buttons *Stresses Corner Nodes*, *Stresses per Element* and *Stresses Gauss Points* (this feature for undeflected structures only).

Z88E, nodal forces calculation. Proceed as follows:
Windows: In the Z88 commander press button *Z88E, Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88E.*

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 axes 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.

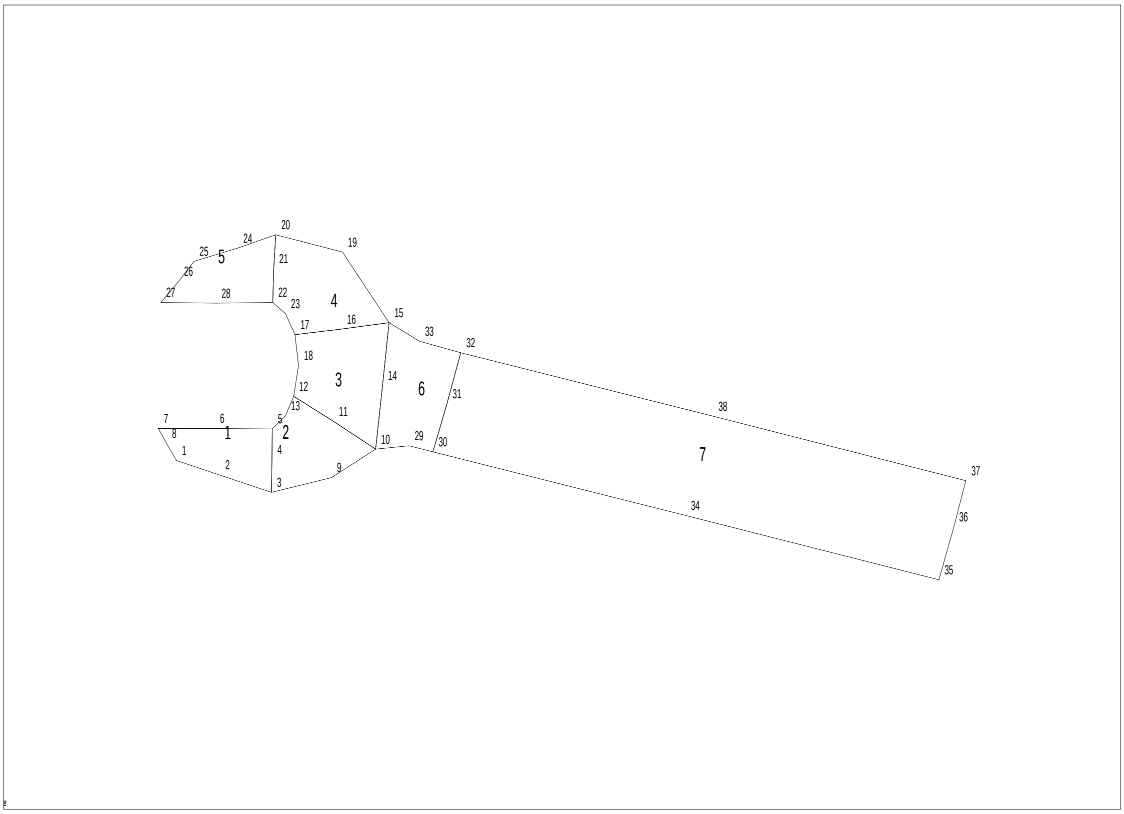
Now calculate this example as indicated above. After that, one can experiment: Subdivide the SE 7 in Z88NI.TXT as a meaningful variation as follows:
7 7 *("Super element 7 is of type 7, i.e. Plane Stress Element No.7")*
6 L 3 E *("Subdivide SE 7 into finite elements Plane Stress No.7 and subdivide into 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 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 Z88O.

With editor:
Write the mesh generator input file Z88NI.TXT (cf. chapter 3.3) with an editor:

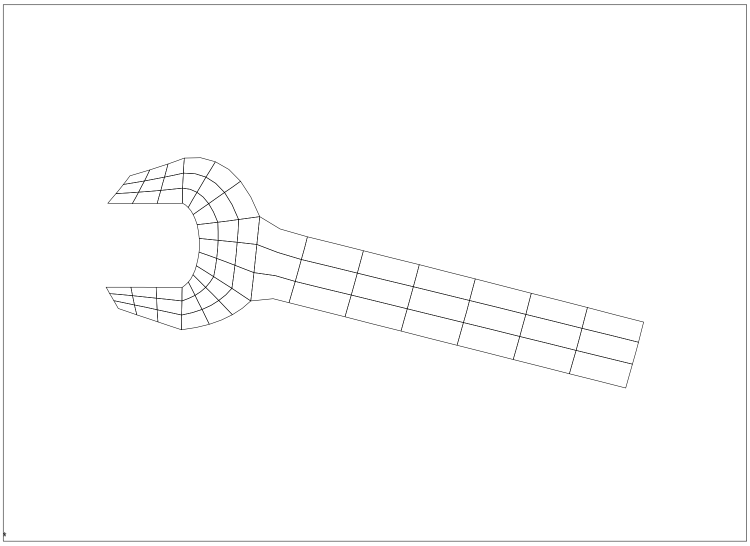
2 38 7 76 1 0 0 0 0 0 *(2-DIM,38 nodes,7SE,76 DOF,1 mat info line, flags 0)*
1 2 22.040 32.175 *(Node 1, 2 DOF, X and Y coordinates)*
2 2 31.913 28.798 *(Node 2, 2 DOF, X and Y coordinates)*
3 2 43.781 24.826
4 2 43.880 32.373
5 2 43.980 39.424

```
 ......                          (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 dropped 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
```

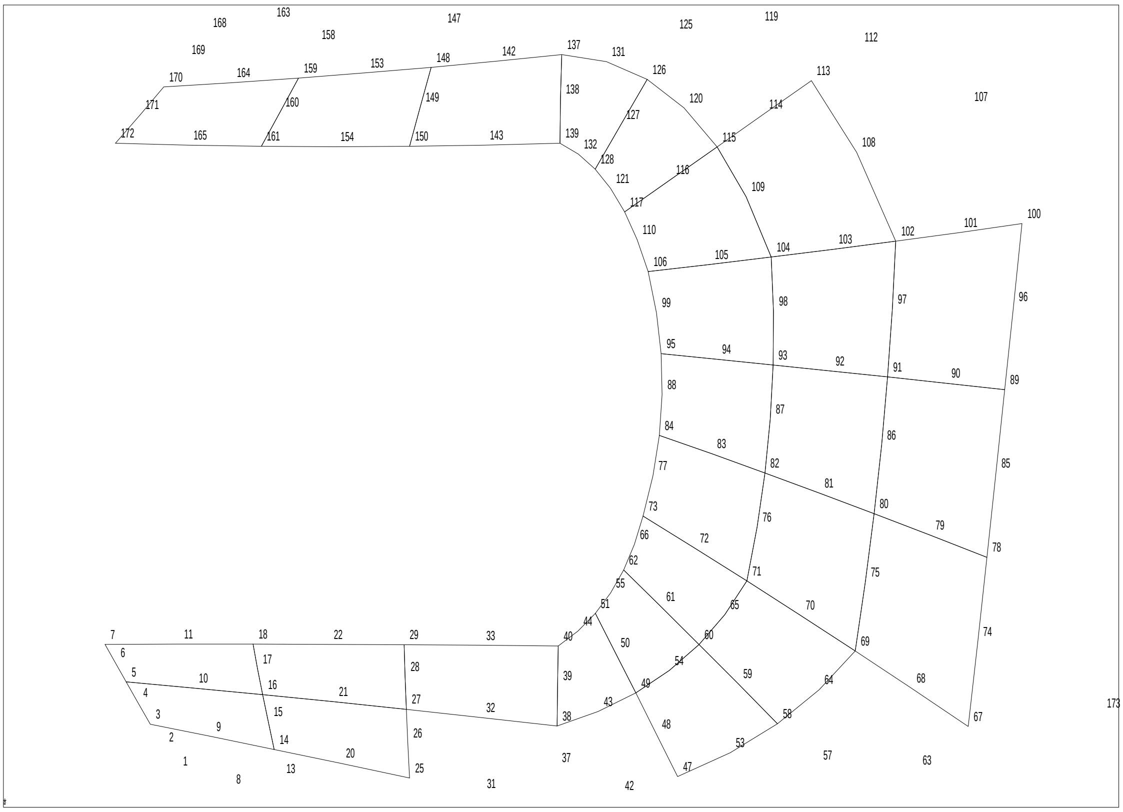
With CAD program and editor:

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) after conversion with Z88X or
- with the Z88 plot program Z88O 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 Z88O.

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   *(3×3 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   *(3×3 Gauss points for stresses, KFLAG 0, von Mises stresses)*

Now launch the Cholesky solver Z88F and then the stress processor Z88D. 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.

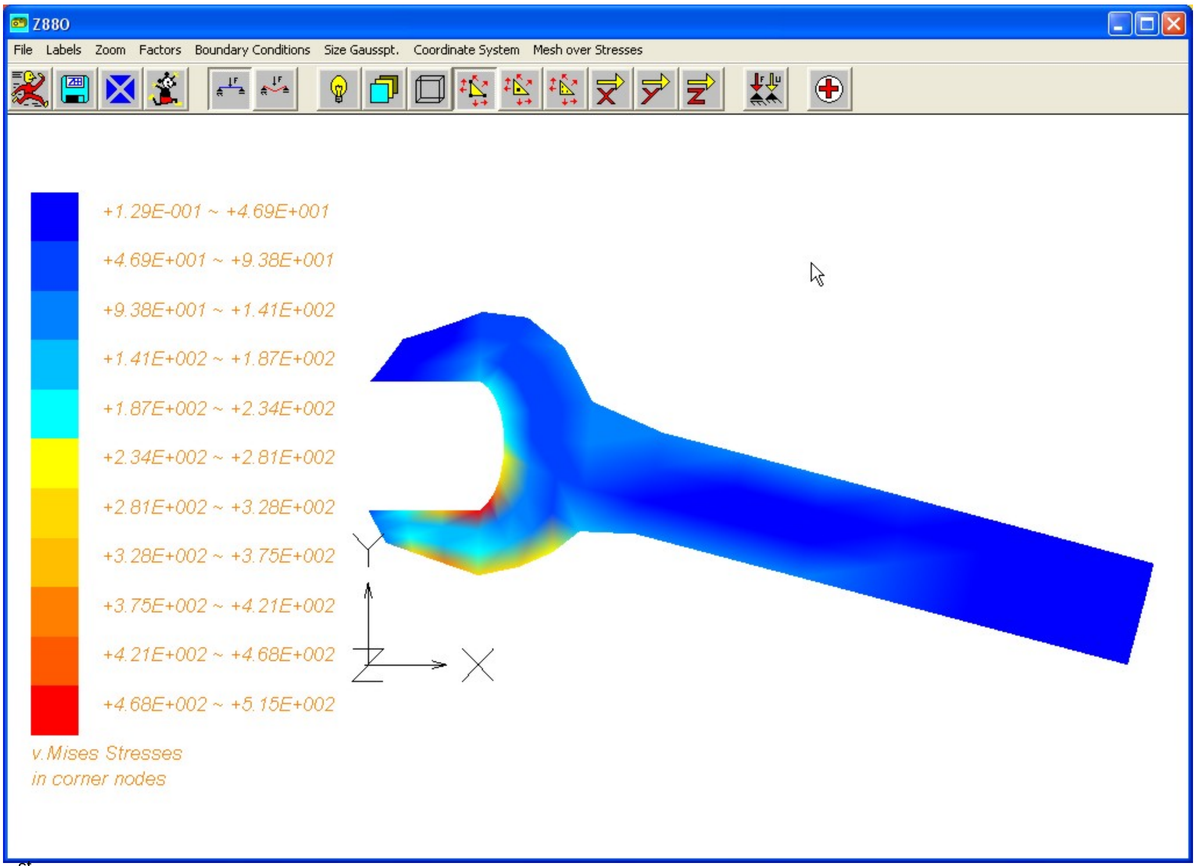
You calculate the nodal forces with Z88E.

5.1.2 Results

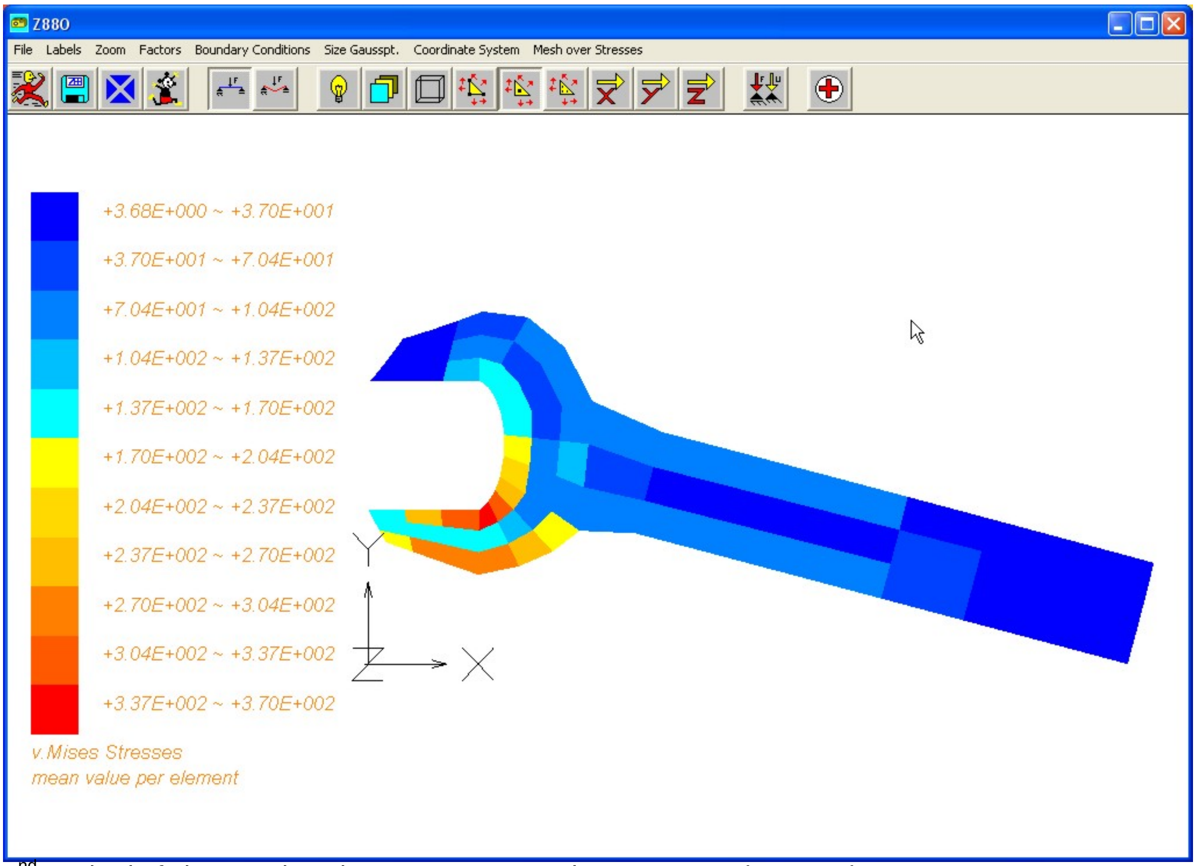
The Cholesky solver Z88F provides the following output files:
Z88O0.TXT stores the processed structure data. It is mainly intended for documentation purposes, but also shows if your input file Z88NI.TXT for the mesh generator did what you meant it to do.
Z88O1.TXT stores the processed boundary conditions: 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 nodal force processor **Z88E** uses internally the calculated deflections of Z88F and stores **Z88O4.TXT**, the computed nodal forces.

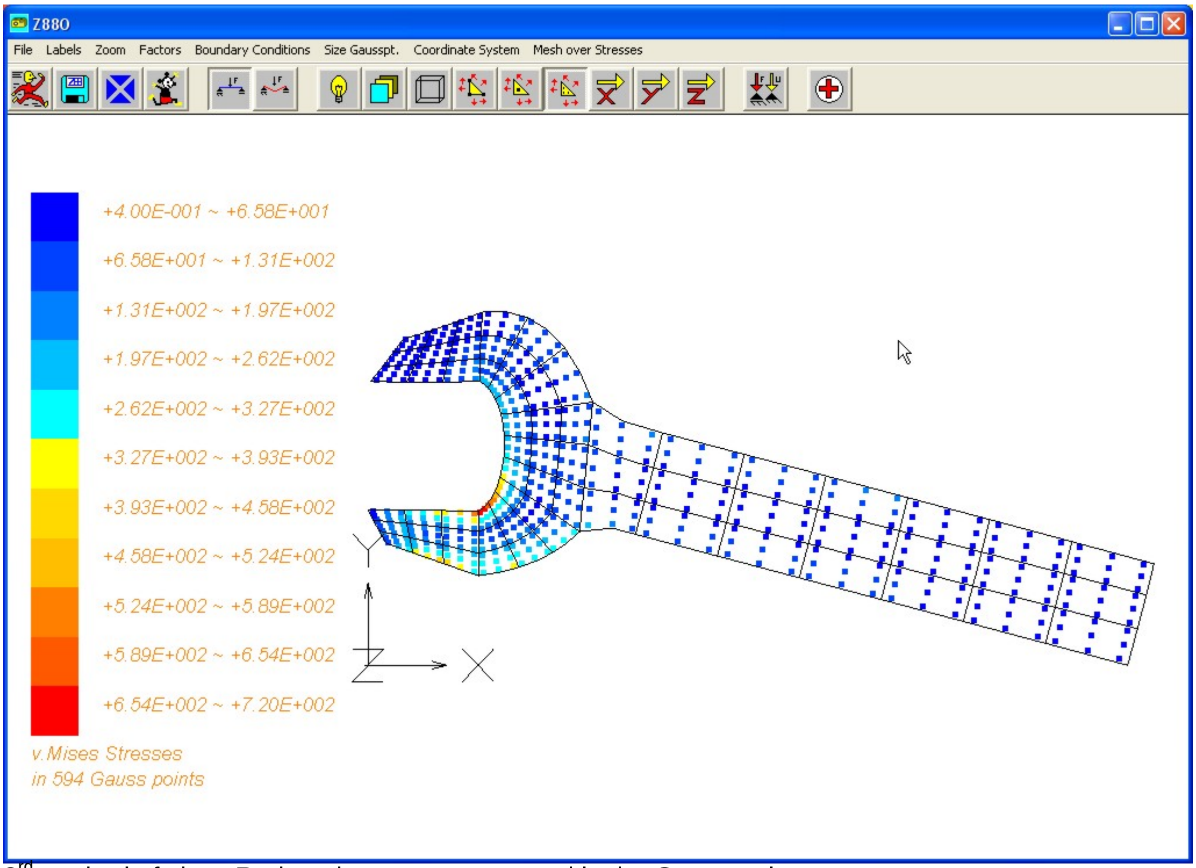
The OpenGL plot program **Z88O** can use three methods of view for reduced stresses: the average reduced stresses per node, average reduced stresses per element and the reduced stresses computed in the Gauss points. These three methods can give very different results depending on stress peaks. Usually the stresses computed in the Gauss points give the highest and most exact values depending on the kind of structure and boundary conditions. Otherwise, if you get nearly the same values for all three methods of view, then your kind of structure and the boundary conditions are very equable.



1st method of view: Reduced stresses in the corner nodes (which are in fact computed from the Gauss points around a node).



2nd method of view: Reduced stresses computed as a mean value per element.



3rd method of view: Reduced stresses computed in the Gauss points.

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: In the Z88 commander launch Z88X, press Button *DXF → Z88I** (default), *Run Button*.
LINUX/UNIX: In the Z88 commander press button *DXF → Z88I** under *CAD converters.*

Z88O, look at finite element structure. Proceed as follows:
Windows: In the Z88 commander launch Z88O, *Run Button*. Use the Wireframe Mode. Rotate the structure with the keys *F2~F7*, reset with *F8*. Pan with the arrow keys and *Home* and *End*, zoom in and out with *Prior* and *Next*. Then press the *Autoscale Button* (3rd button from left), to reset all transformations. Switch on *Mouse* (4th button from left). Now you may zoom by the left mouse key pressed, pan by the middle mouse key pressed and rotate by the right mouse key pressed. You may also label the nodes or elements: *Menu* > *Labels* etc.
LINUX/UNIX: In the Z88 commander launch Z88O, *Run Button*. Use the Wireframe Mode. Rotate the structure with the keys *F2~F7*, reset with *F8*. Pan with the arrow keys and *Home* and *End*, zoom in and out with *Prior* and *Next*. Then press the *Autoscale Button* (6th button from top), to reset all transformations. Switch on *Mouse* (7th button from top). Now you may zoom by the left mouse key pressed, pan by the middle mouse key pressed and rotate by the right mouse key pressed. You may also label the nodes or elements: *Menu* > *Labels* etc.

Z88F, calculates displacements. Proceed as follows:
Windows: In the Z88 commander press button *Z88F, CD Button* (is default), *Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88F -C*.

Z88D, calculates stresses. Proceed as follows:
Windows: In the Z88 commander press button *Z88D, Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88D*.

Z88E, nodal forces calculation. Proceed as follows:
Windows: In the Z88 commander press button *Z88E, Run Button.*

LINUX/UNIX: In the Z88 commander press button *Z88E*.

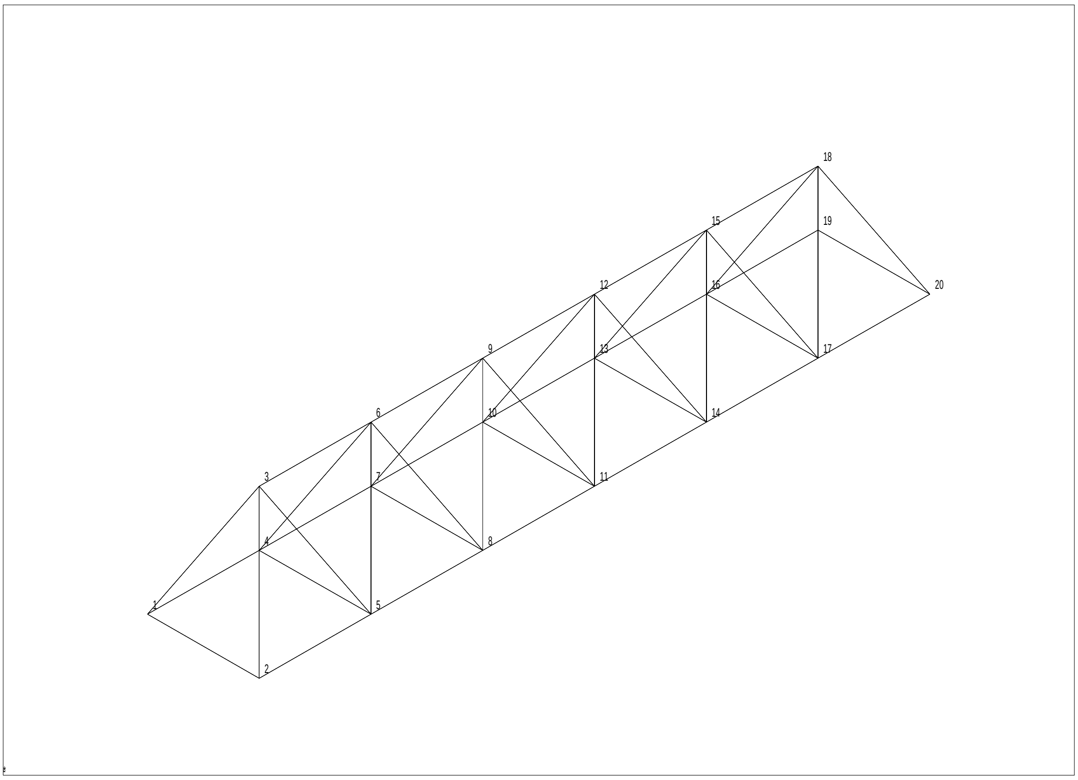
Z88O, Finite Elemente Struktur verformt betrachten. Vorgehen:
Windows, LINUX/UNIX: look at the deflected finite element structure. The deflections are magnified per default by the factor 100 which is correct for this example. If you'll press the three displacement buttons (X, Y and Z) you'll notice that the nodal displacements are plotted with different colours. Compare this to the values in the displacement file Z88O2.TXT. If you

enter *0 0 1* in the flag file for stresses Z88I3.TXT and run Z88D once more, then firing up again Z88O you’ll see the tensile stresses as “reduced stresses” by switching to *Reduced stresses mean value per Element* (Windows) or *Stresses per Element* (UNIX).

The example is simple and straight. Experiment with the 3D features of the plot program 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^2, 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 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 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 *(Node 1, DOF 2, Displacement 0 (=fixed in Y direction)*
1 3 2 0 *(Node 1, DOF 3, Displacement 0 (=fixed in Z direction)*
2 1 2 0 *(Node 2, DOF 1, Displacement 0 (=fixed in X direction)*
2 3 2 0 *(Node 2, DOF 3, Displacement 0 (=fixed in Z direction)*
7 3 1 -30000 *(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.

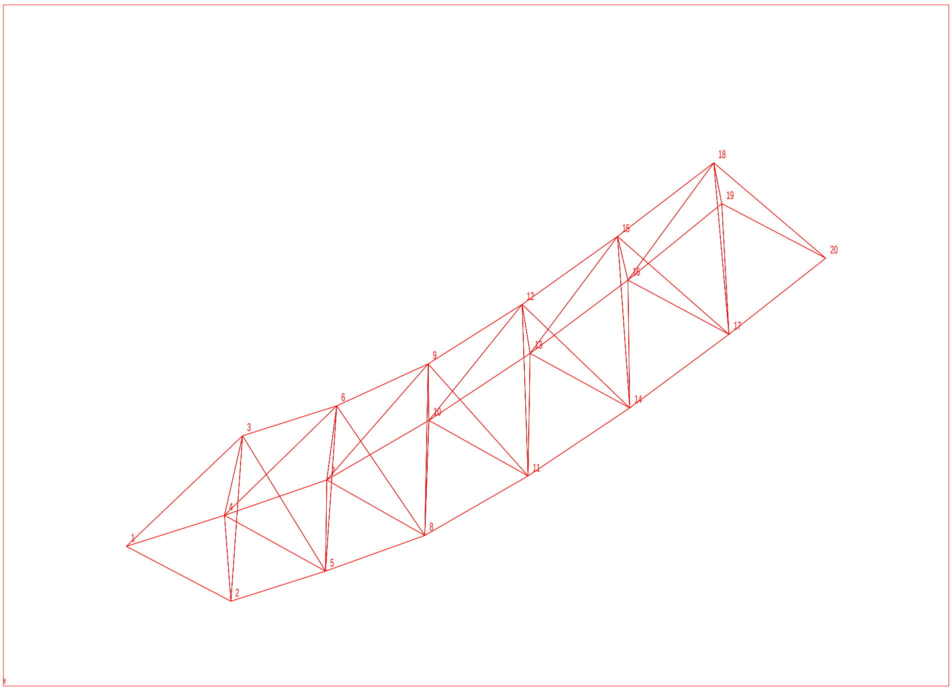
CAD and editor:
Because now the structure data Z88I1.TXT, the boundary conditions Z88I2.TXT and the header file for the stress processor Z88I3.TXT (with any content) do exist, you can launch

- Z88F Cholesky solver for computing the deflections
- Z88D stress processor
- Z88E nodal force processor

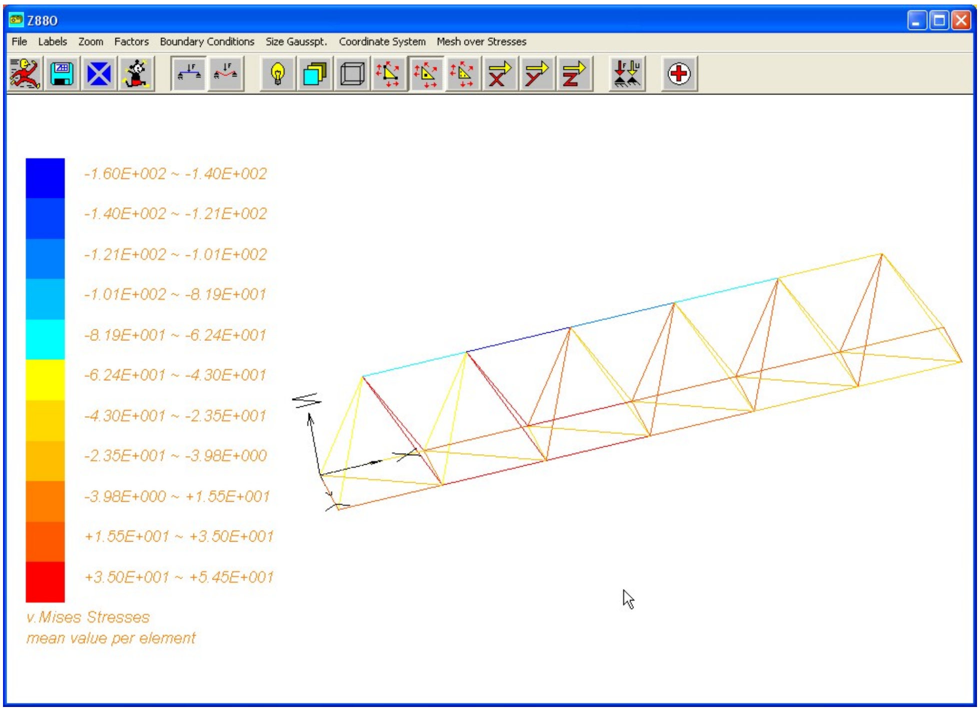
5.2.2 Results

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):



Reduced stresses are not provided in the plot program Z88O for Trusses No.4. But Z88O does it because tensile stresses of trusses are equivalent to *von Mises* stresses. Why don't you try it? You only have to outwit Z88O by entering 0 0 1 into Z88I3.TXT. Run Z88D. Then launch Z88O and switch to *Reduced stresses mean values per element*. You should switch off *Mesh over stresses*.



Plotting tensile stresses with Z88O. Enter 0 0 1 into Z88I3.TXT, run Z88D again.

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: In the Z88 commander launch Z88X, press Button *DXF → Z88I** (default), *Run Button*.
LINUX/UNIX: In the Z88 commander press button *DXF → Z88I** under *CAD converters*.

Z88O, look at finite element structure. Proceed as follows:
Windows: In the Z88 commander launch Z88O, *Run Button*. Use the Wireframe Mode. Switch on *Mouse* (4th button from left). You may also label the nodes and elements: *Menu* > *Labels* > *Label All*.
LINUX/UNIX: In the Z88 commander launch Z88O, *Run Button*. Use the Wireframe Mode. Switch on *Mouse* (7th button from top). You may also label the nodes and elements: *Menu* > *Labels* > *Label All*.

Z88F, calculates displacements. Proceed as follows:
Windows: In the Z88 commander press button *Z88F*, *CD Button* (is default), *Run Button*.

LINUX/UNIX: In the Z88 commander press button *Z88F -C*.

Z88D, calculates stresses. Proceed as follows:
Windows: In the Z88 commander press button *Z88D, Run Button*.

LINUX/UNIX: In the Z88 commander press button *Z88D*.

Z88E, nodal forces calculation. Proceed as follows:
Windows: In the Z88 commander press button *Z88E*, *Run Button*.

LINUX/UNIX: In the Z88 commander press button *Z88E*.

Z88O, look at the deflected finite element structure. Proceed as follows:
Windows, LINUX/UNIX: Choose *Deflected*. The deflections are magnified per default by the factor 100 which is not enough for this example. Enter 1,000 each for *FUX*, *FUY* and *FUZ* in *Menu* > *Factors* > *Deflections*.

Basically, the calculation and displaying of reduced 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 with beam or cam elements; however, plane stress elements like No.7 would do this job, of course) 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

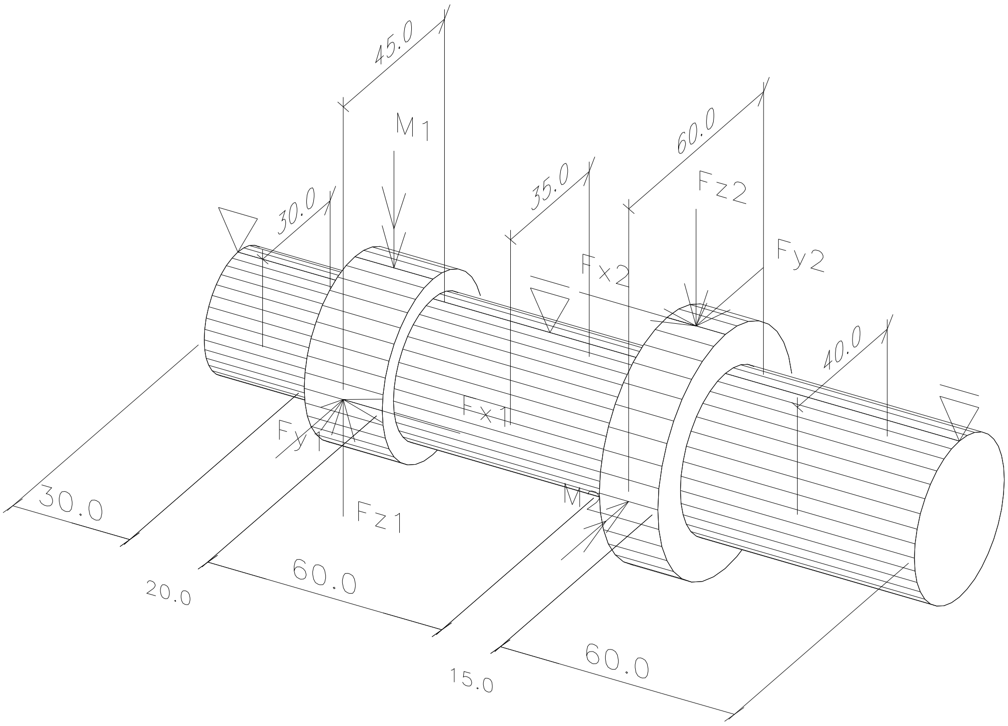
For the loads we picture the cam with the following coordinate system: If we look onto the cam as the main view, then the origin should be at the left end in the middle of the cam. X runs along the cam, Z runs to the upper direction, Y runs in the rear.

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 under-defined !



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 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   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                 (Node 1, DOF 1 (=X) fixed)
  1   2   2        0                 (Node 1, DOF 2 (=Y) fixed)
  1   3   2        0                 (Node 1, DOF 3 (=Z) fixed)
  1   4   2        0                 (Node 1, DOF 4 (=torsion) fixed)
  3   1   1   -10801                 (Node 3, DOF 1 (=X), load -10,801 N)
  3   2   1    +6809                 (Node 3, DOF 2 (=Y), load 6,809 N)
  3   3   1   +18708                 (Node 3, DOF 3 (=Z), load 18,708 N)
  3   4   1  -420930                 (Node 3, DOF 4 (torsion) -420,930 Nmm)
  3   6   1  -243023                 (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

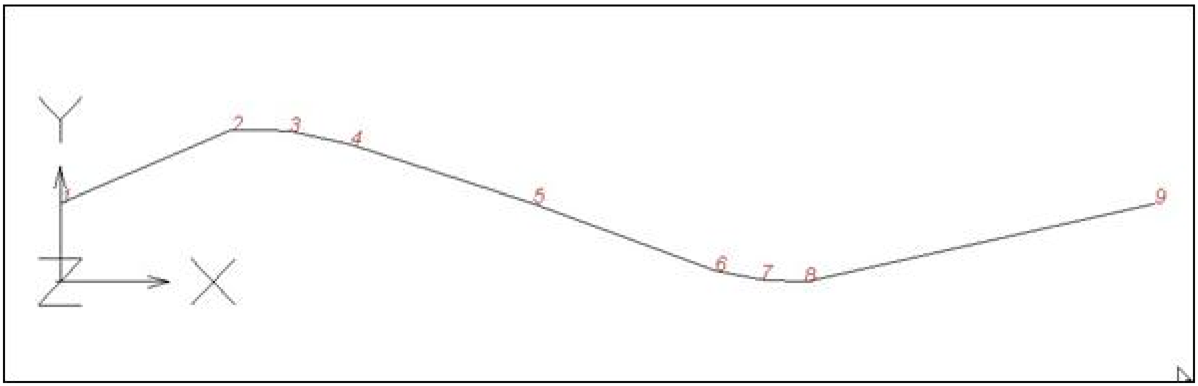
• Z88F	Cholesky solver for computing the deflections
• Z88D	stress processor
• Z88E	nodal force processor

5.3.2 Results

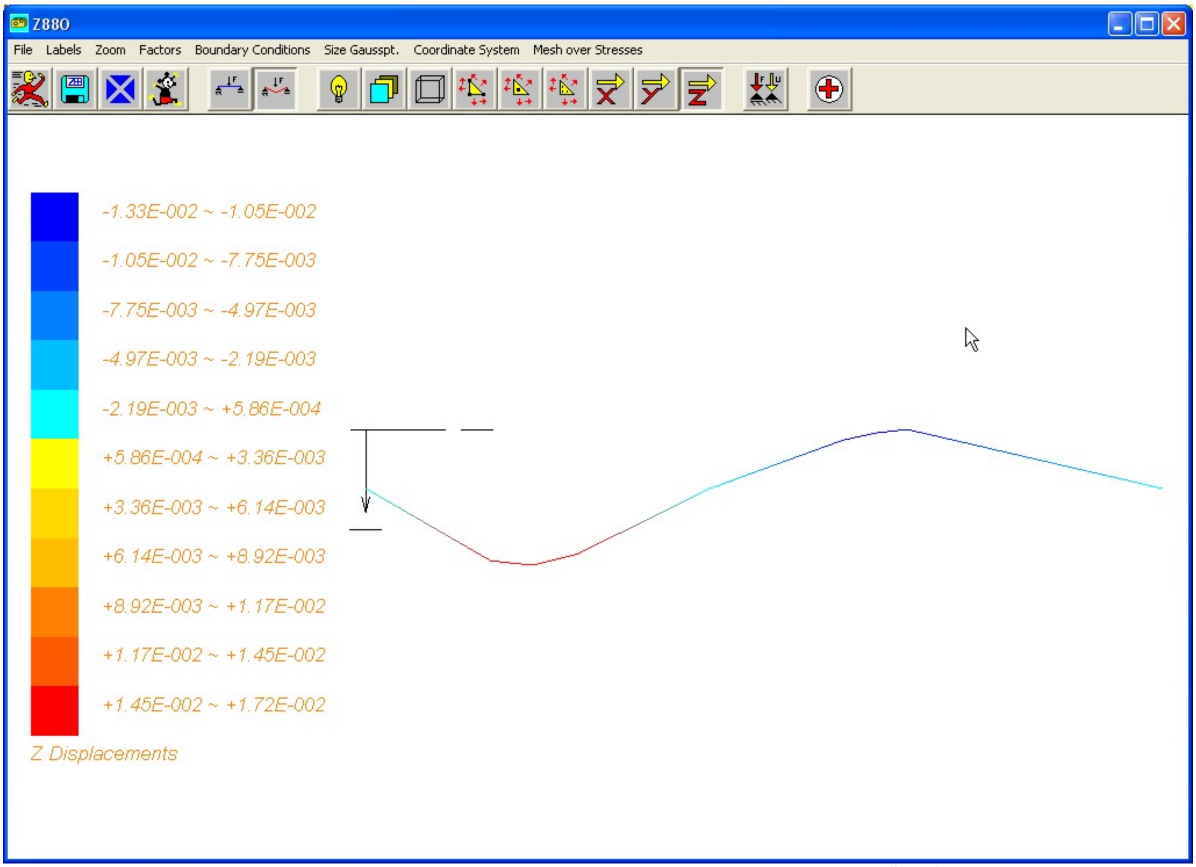
The Cholesky solver Z88F provides the following output files:

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 XY pane, deflected structure with nodal labels



View of XZ pane, deflected with coarse plot of the Z deflections (*Button Deflections Z*). Compare this to the exact values in Z88O2.TXT.

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"
Z88O, looking at structure, structure file Z88I1.TXT
Z88F calculates deflections
Z88D calculates stresses
Z88E calculates nodal forces
Z88O, 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 ℓ= 1,000 mm, cross-cut 50 x 10 mm. Thus: A = 500 mm^{2,} I_{zz} = 4,167 mm^4, e_{zz} = 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:

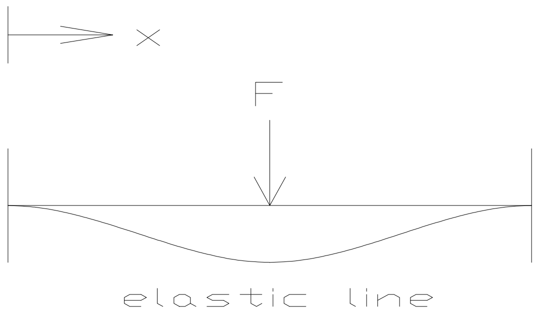
f in the middle: $f = \frac{F\ell^3}{192\,EI} = 10\,\text{mm}$

f in the inflection points: $f_w = \frac{f}{2} = 5\,\text{mm}$

The bending moments on the left, middle, on the right: $M_b = \frac{F\ell}{8} = 206.000\,\text{Nmm}$

The slope angle in the inflection points: $\psi = \operatorname{atan}\left(\frac{3f}{\ell}\right) = 0{,}029991\,\text{rad}$

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 0	*(2-D, 5 nodes, 4 ele, 5 DOF, 1 mat info, KFLAG 0, IBFLAG 1, IPFLAG 0, IQFLAG 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.

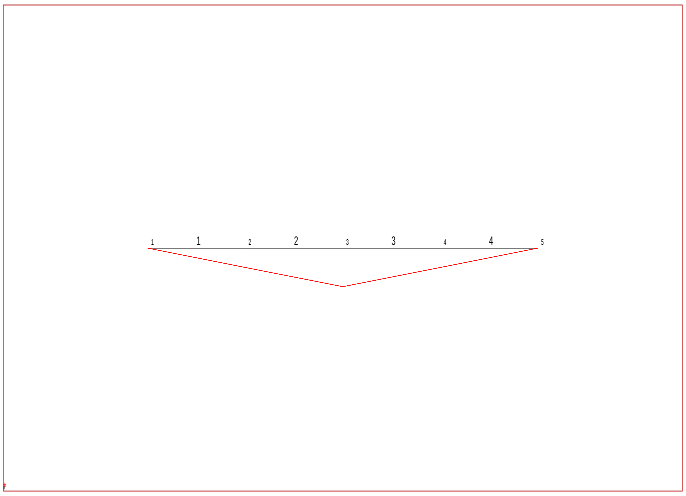
5.4.2 Results

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.

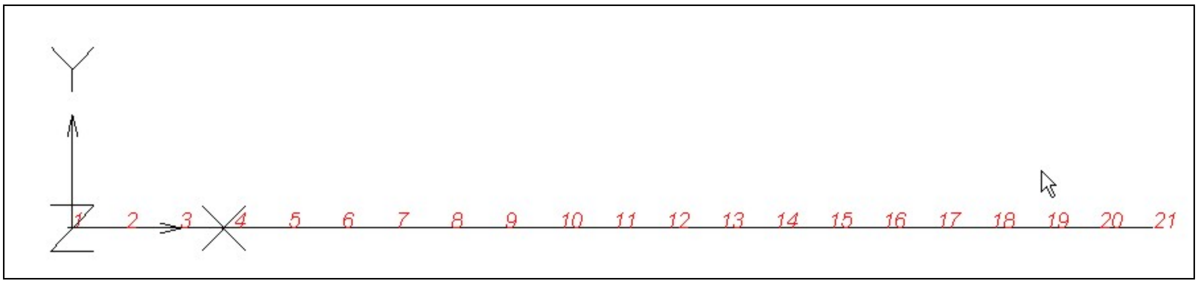


View of undeflected and deflected structure

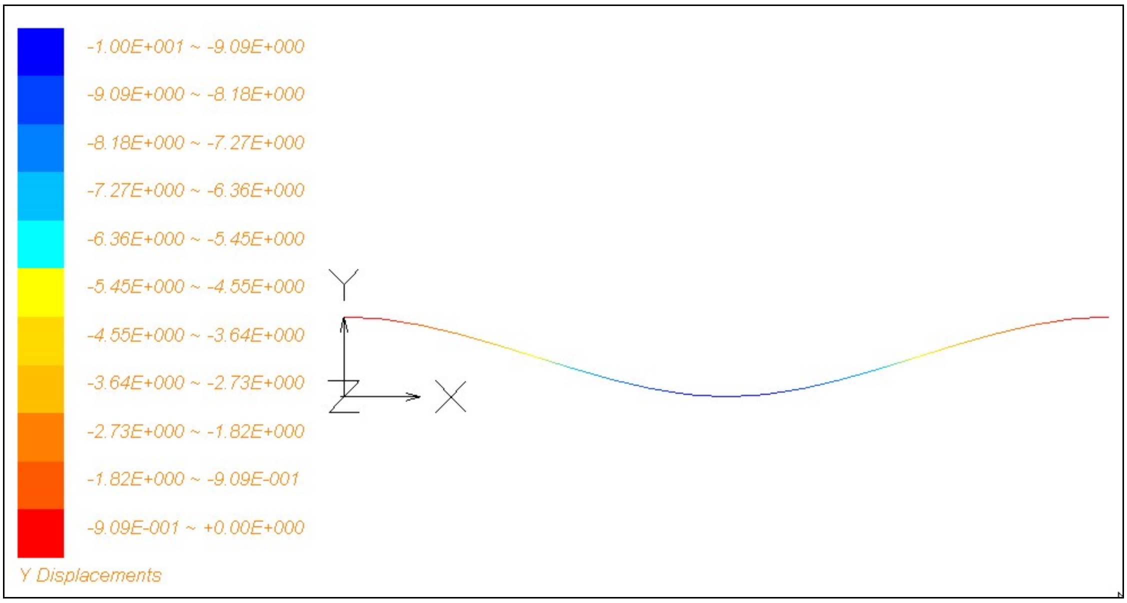
Take into account: The plot program Z88O connects the nodes with straight lines, although the deflection curve represents a cubic parable in the case of a Beam No.13 or No.2. This means: Z88O shows the deformations correctly for the node, but straight lines are between the nodes. Therefore, no deflection curve is shown. If you want to plot a real nice deflection curve with Z88O, 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). Just do it: Erlarge thy file Z88I1.TXT to 21 nodes and 20 elements und modify the boundary conditions file as follows:

```
  6
  1    1    2   0.
  1    2    2   0.
  1    3    2   0.
 11     2    1 -1648.
 21     2    2  0.
 21     3    2  0.
```

Of course, this was prepared for you: B4_20ELE_X.DXF → Z88X.DXF, then run Z88X and compute the structure.



New structure with 21 nodes and 20 elements



New structure with 21 nodes and 20 elements

As you see we've got a quite nice line of deflection. And we may also plot the values of the deflections. Factors for Deflection FUX, FUY and FUZ 10 each.

Keep in mind: Place nodes into your structure where you want to get results!

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"
Z88O, looking at super structure, super structure file Z88NI.TXT
Z88N, computes the finite element mesh
Z88O, 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.

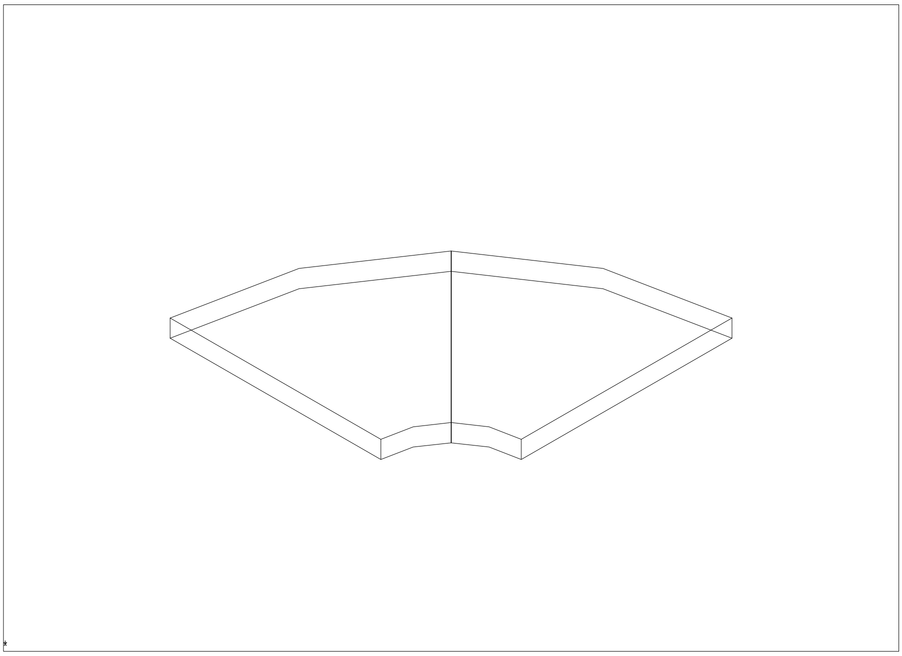
Z88: (in reduced form, more detailed instructions cf. examples 5.1, 5.2 and 5.3)
Z88X, conversion, "from Z88X.DXF to Z88I*.TXT"
Z88F calculates deflections

Z88D calculates stresses
Z88O plots FE structure, now 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 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 Z88O).

With editor:

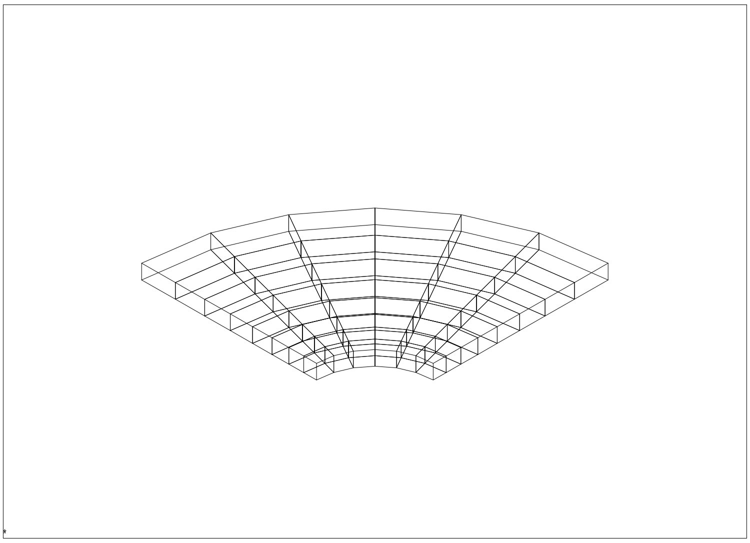
Write the mesh generator input file Z88NI.TXT (cf. chapter 3.3) with an editor:

```
 3   32   2   96   1   1   0   0   0   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 Z88O 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 *(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 *(Node 65, DOF 1 fixed)*
65 2 2 0 *(Node 65, DOF 2 fixed)*
65 3 2 0 *(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):

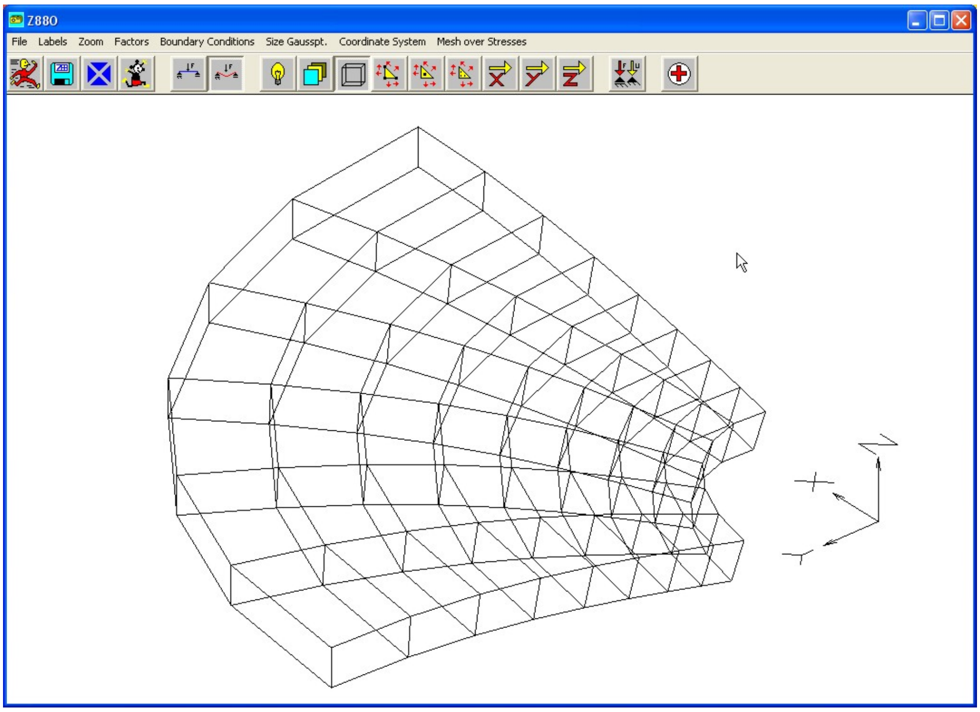
2 0 1 *(2×2 Gauss points for stresses, KFLAG 0, von Mises stresses)*

CAD and editor:
Now launch the Cholesky solver Z88F and then the stress processor Z88D. Compute nodal forces with Z88E.

5.5.2 Results

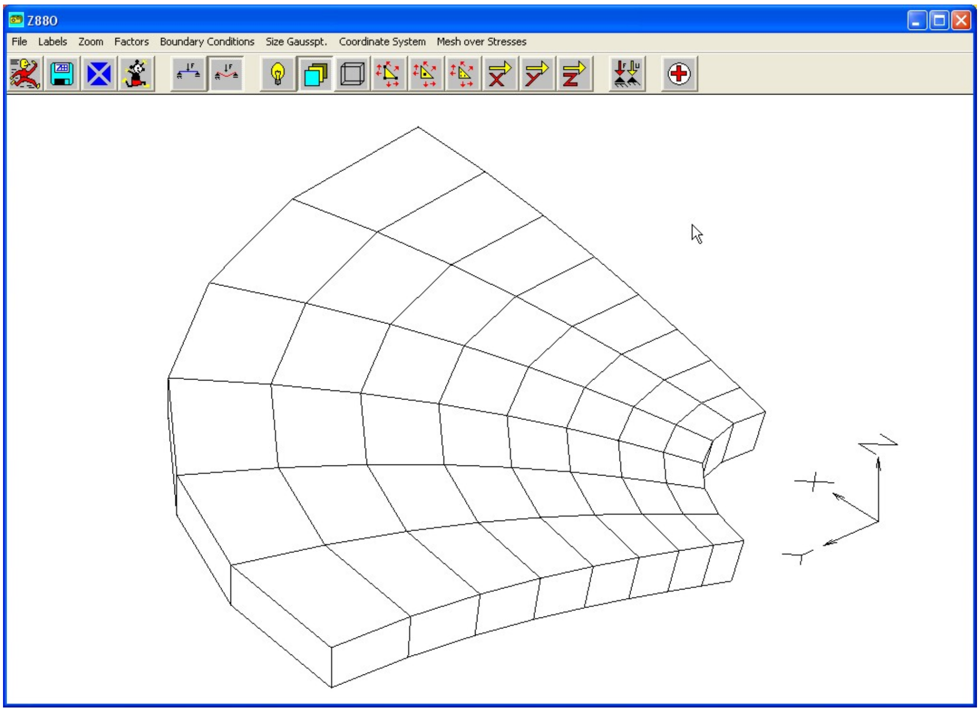
The Cholesky solver Z88F provides the following output files:
Z88O0.TXT stores the processed structure data. For documentation purposes. **Z88O1.TXT** stores the processed boundary conditions: For documentation purposes. **Z88O2.TXT**, the displacements, the main task and solution of the FEA problem. 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):



View of the deflected structure, Wireframe Mode.

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!



View of the deflected structure, Hiddenline Mode.

Hint: In reality you won’t compute such a structure with hexahedrons with linear shape functions (Type No.1) but with hexahedrons with quadratic shape functions (Type No.10). See *Rieg, F.; Hackenschmidt, R.: Finite Elemente Analyse für Ingenieure. 3. Auflage. München Wien. Carl Hanser: 2009 (in German language).*

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"
Z88O, looking at structure, structure file Z88I1.TXT
Z88F calculates deflections
Z88D calculates stresses
Z88E calculates nodal forces
Z88O, 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^2). 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):

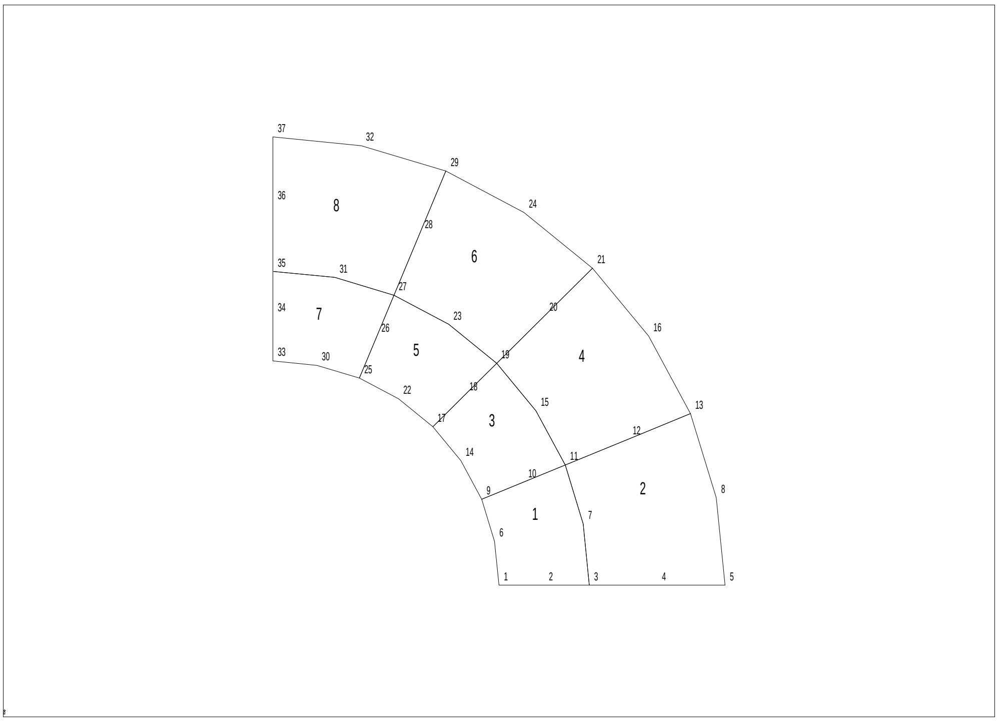
"1/6 points": 10,472 N
"2/3 points": 41,888 N
"2/6 points": 20,944 N

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 φ = 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 0 *(2D, 37 nodes, 8 ele, 74 DOF, 1 mat info, polar coor., other flags 0)*
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  0     ( 2D, 37 nodes, 8 elements, 74 DOF, 1 mat info, Polar coor.,
                                 beam & plate & surface loads 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       (Node 1, DOF 1(= X), a load of 10,472 N)
  1  2  2  0           (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 *(3×3 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

Boundary conditions by edge loads:

The data entry by single forces was somewhat cumbersome because of dividing the force of 251.327 N to several nodal points with respect to the actual angle position. It is much more easier to enter edge loads by the surface and pressure loads file Z88I5.TXT. The edge load is:

$$q = \frac{F}{\ell} = \frac{F}{r \cdot \varphi} = \frac{251327}{40 \cdot \pi/2} = 4000\, N/mm$$

This edge load acts onto the elements 1, 3, 5 and 7. The edge of element 1 is the edge defined by the corner nodes 9 and 1 and the middle node 6 etc. The edge load points normally to the edge, there are no tangential loads. Thus, the surface and pressure loads file Z88I5.TXT is:

```
4
1  4000.  0.   9   1   6
3  4000.  0.  17   9  14
5  4000.  0.  25  17  22
7  4000.  0.  33  25  30
```

To make the solvers reading in the surface and pressure loads file you are to set the surface and pressure loads flag to 1 in the first line of the general data file Z88I1.TXT:

2 37 8 74 1 0 0 0 **1**
↑ surface and pressure loads flag IQFLAG

Now edit the boundary conditions file Z88I2.TXT: Skip all forces:

```
10
 1   2   2   0.
 2   2   2   0.
 3   2   2   0.
 4   2   2   0.
 5   2   2   0.
33   1   2   0.
34   1   2   0.
35   1   2   0.
36   1   2   0.
37   1   2   0.
```

Please see the sample files B6_Q_1.TXT, B6_Q_2.TXT and B6_Q_5.TXT on the CD-ROM or the Internet packages. Now compute deflections, stresses and nodal forces as usual:

- 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.

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. Now experiment.. you have 5 × 2 × 2 = 20 possibilities.

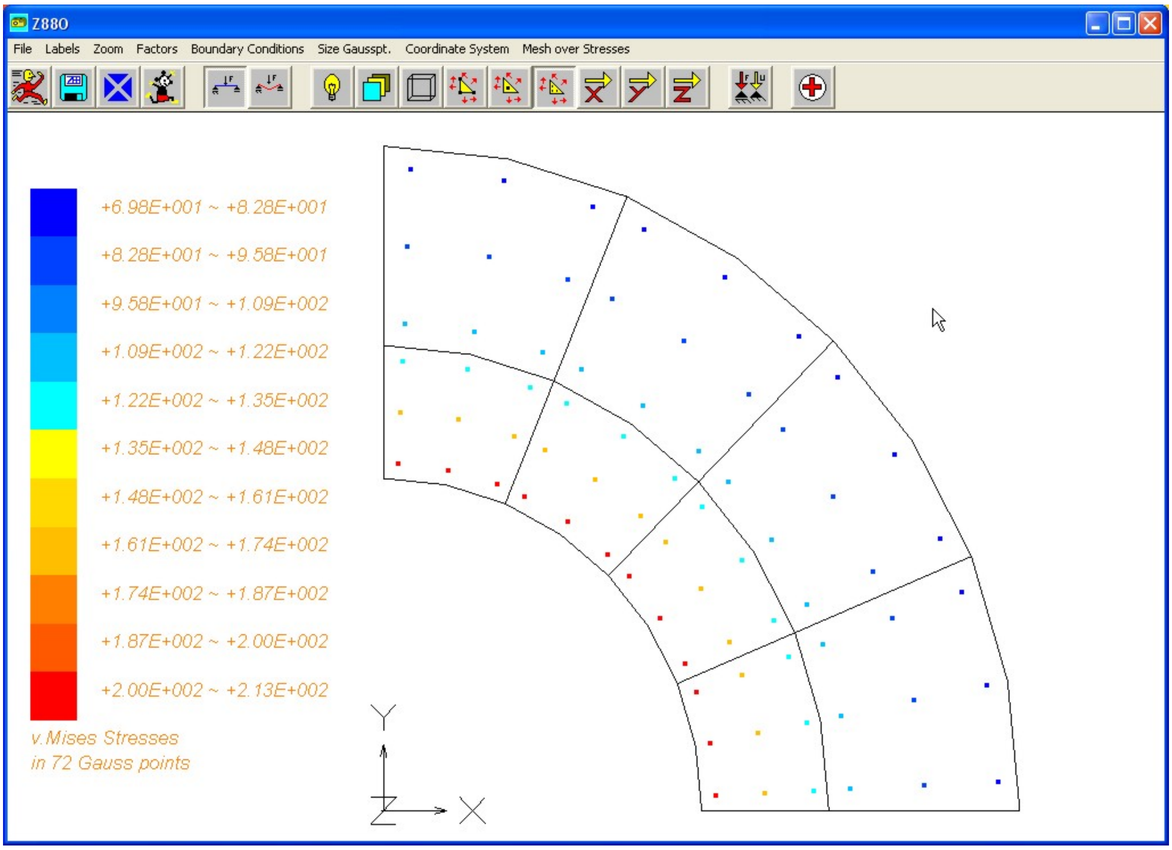
Now you should plot the boundary conditions. The problem here is that some nodes have more than one BC e.g. node 1 features a force in X direction and is fixed in Y direction, too:

```
 1    1    1   +1.04720E+004
 1    2    2   +0.00000E+000
```

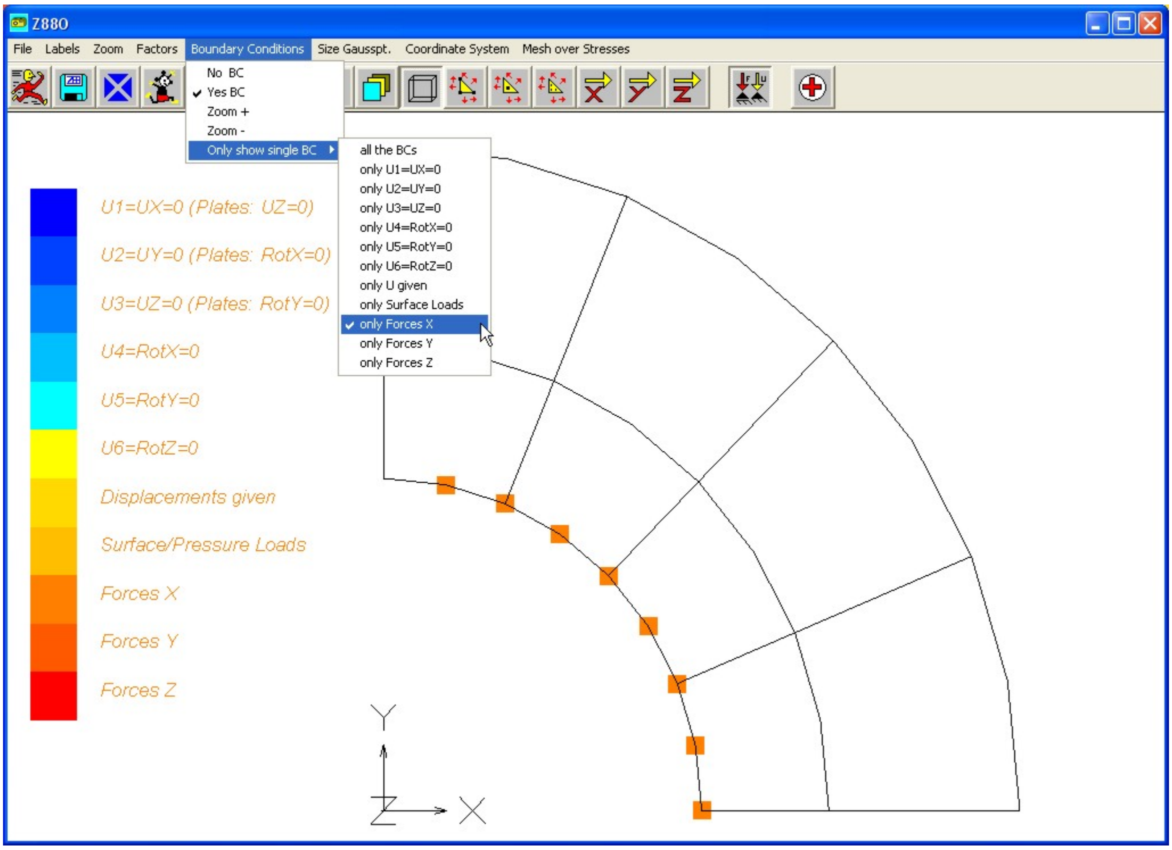
Thus, you select the BC you want to see.
Suppose you would now choose a hardened steel. This means to compute the *principal stresses* not the *von Mises* stresses. Thus, the third entry in Z88I3.TXT would now read 2:

```
3    1    2
```

Then do a stress calculation again with Z88D.



Plot of the stresses in the Gauss points with Z88O



Example for the selective plot of BCs with Z88O

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 super structure. 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"
Z88O, structure file Z88NI, look at the super structure
Z88N, mesh generator, produces Z88I1.TXT
Z88O, structure file Z88I1.TXT, undeflected 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
Z88O, plots FE structure, now also deflected 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 = r_d (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 (assuming Young's modulus E= 206000 N/mm^2 and Poisson's ratio υ= 0,3):

pressure:

$$p = \frac{r_d E}{r_i} \cdot \frac{1}{\frac{1+q_a}{1-q_a}+\nu} = 262 \text{ N/mm}^2 = 2.620 \text{ bar} \quad \textit{with} \quad q_a = \frac{r_i^2}{r_a^2} = 0,25$$

radial stresses:

$$\sigma_{r_i} = -p = -262 \text{ N/mm}^2$$

$$\sigma_{r_a} = 0$$

tangential stresses:

$$\sigma_{t_i} = p \cdot \frac{1+q_a}{1-q_a} = 437 \text{ N/mm}^2$$

$$\sigma_{t_a} = 2p \cdot \frac{q_a}{1-q_a} = 175 \text{ N/mm}^2$$

von Mises stresses: $\sigma_{Vi} = \sqrt{\sigma_r^2 + \sigma_\theta^2 - \sigma_r \cdot \sigma_\theta} =$

$$\sqrt{(-262)^2 + 437^2 - (-262) \cdot 437} = 612 \text{ N/mm}^2$$

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: $F = p\,A = p\,2\,\pi\,r_i\,\ell = 2.633.911 \text{ N}$
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.

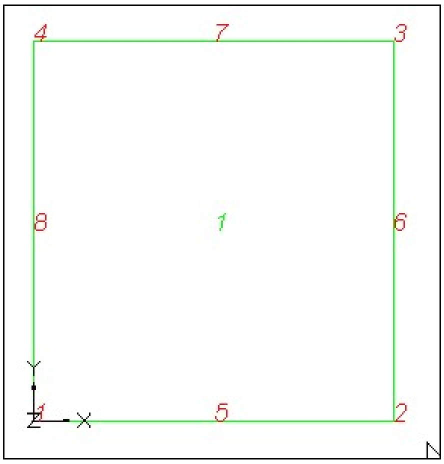
Make sure that cylindrical coordinates are always expected for Torus No.6, No.8 and No.12, i.e. radius R (replaces X) and height coordinate Z (replaces Y). R and Z must feature always positive values ! KFLAG must be zero!

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 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 Z88O:



Super structure Z88NI.TXT

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  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 Z88O 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    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 *(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:

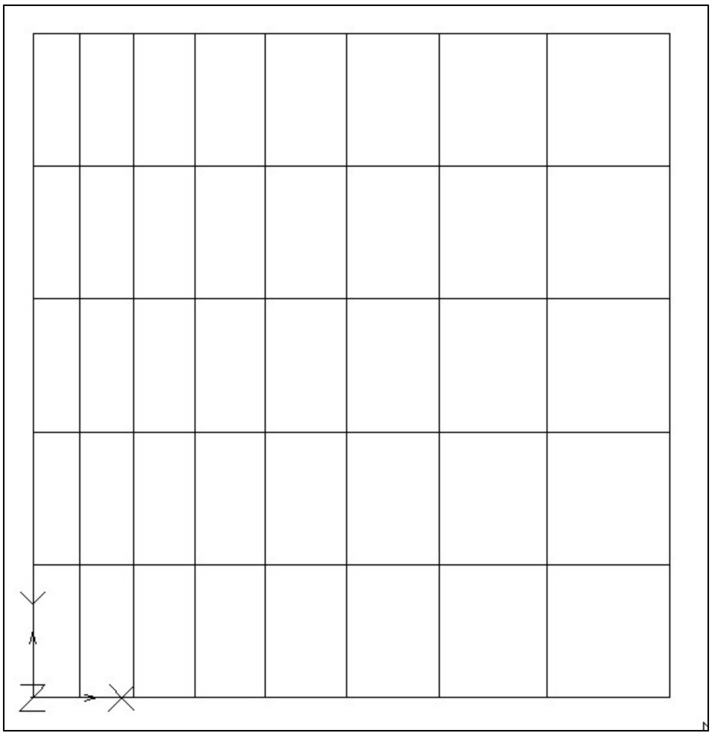
In the CAD program:
Switch to the layer Z88GEN and write with the TEXT function into any free place:

Z88I3.TXT 3 0 1 *(3 × 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

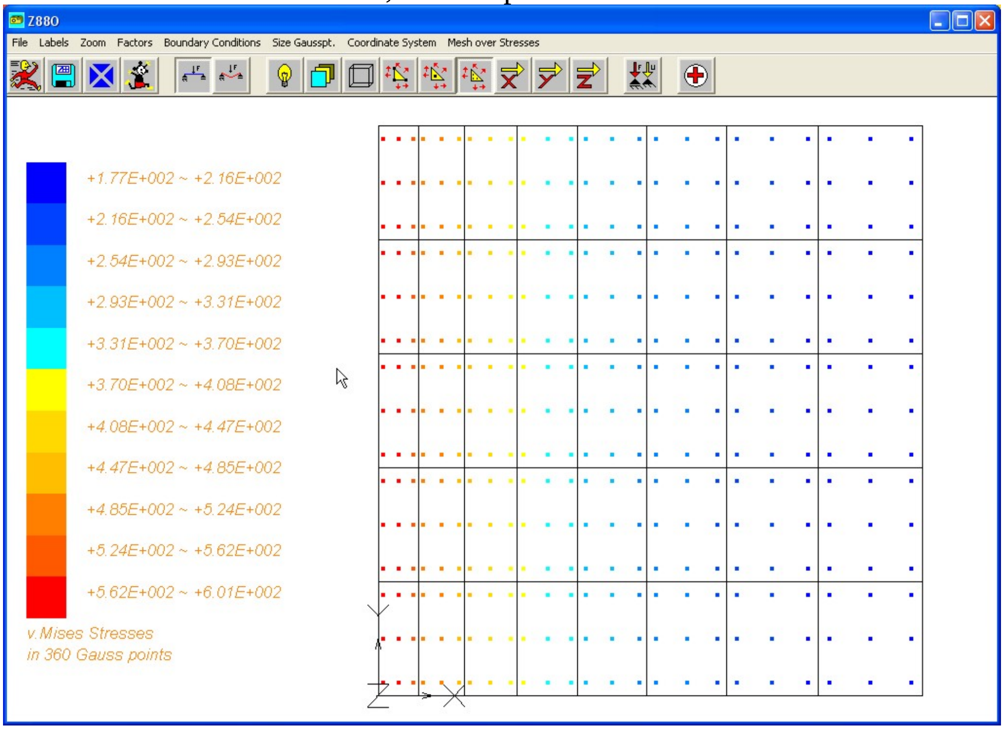
CAD and editor:

Now launch the Cholesky solver Z88F and then the stress processor Z88D. 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

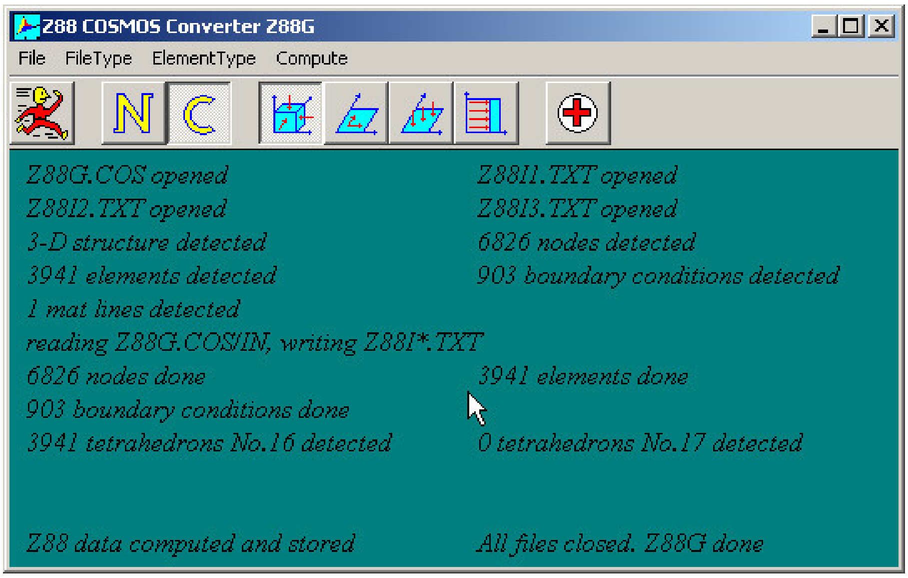
Copy the sample file B11_G.COS to the Z88 input file Z88G.COS.

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 centre 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 axes. The flange facings of the shaft axes 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*

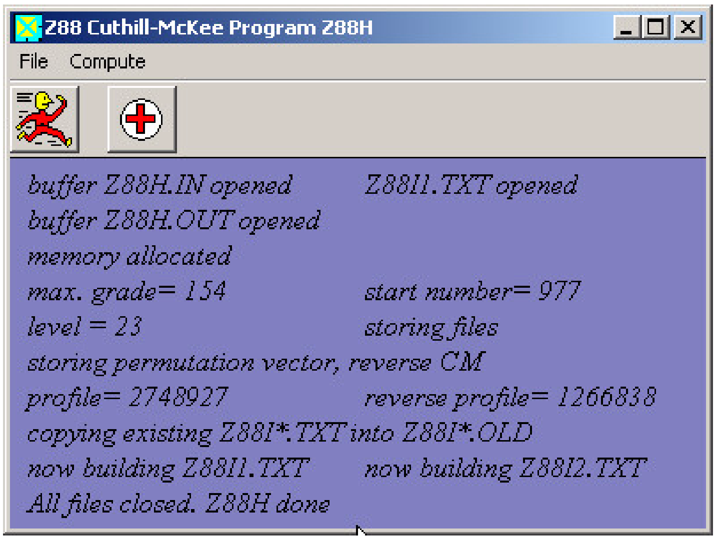
Start converting Z88G.COS with **Z88G**



```
Z88G.COS opened                         Z88I1.TXT opened
Z88I2.TXT opened                        Z88I3.TXT opened
3-D structure detected                  6826 nodes detected
3941 elements detected                  903 boundary conditions detected
1 mat lines detected
reading Z88G.COS/IN, writing Z88I*.TXT
6826 nodes done                         3941 elements done
903 boundary conditions done
3941 tetrahedrons No.16 detected        0 tetrahedrons No.17 detected


Z88 data computed and stored            All files closed. Z88G done
```

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.



```
buffer Z88H.IN opened          Z88I1.TXT opened
buffer Z88H.OUT opened
memory allocated
max. grade= 154                start number= 977
level = 23                     storing files
storing permutation vector, reverse CM
profile= 2748927               reverse profile= 1266838
copying existing Z88I*.TXT into Z88I*.OLD
now building Z88I1.TXT         now building Z88I2.TXT
All files closed. Z88H done
```

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.

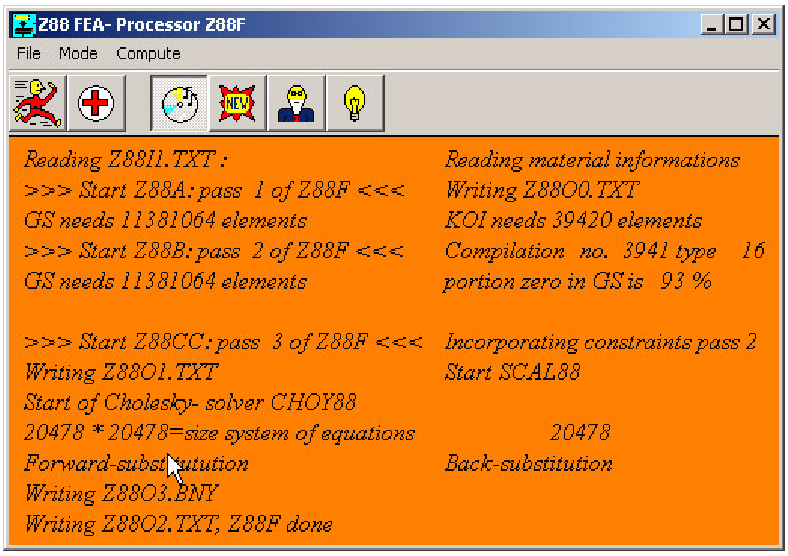
Thus, **Z88.DYN** should look as follows:

MAXGS	when starting, any value
MAXKOI	minimum 39,410

MAXK minimum 6,826
MAXE minimum 3,941
MAXNFG minimum 20,478
MAXNEG minimum 1

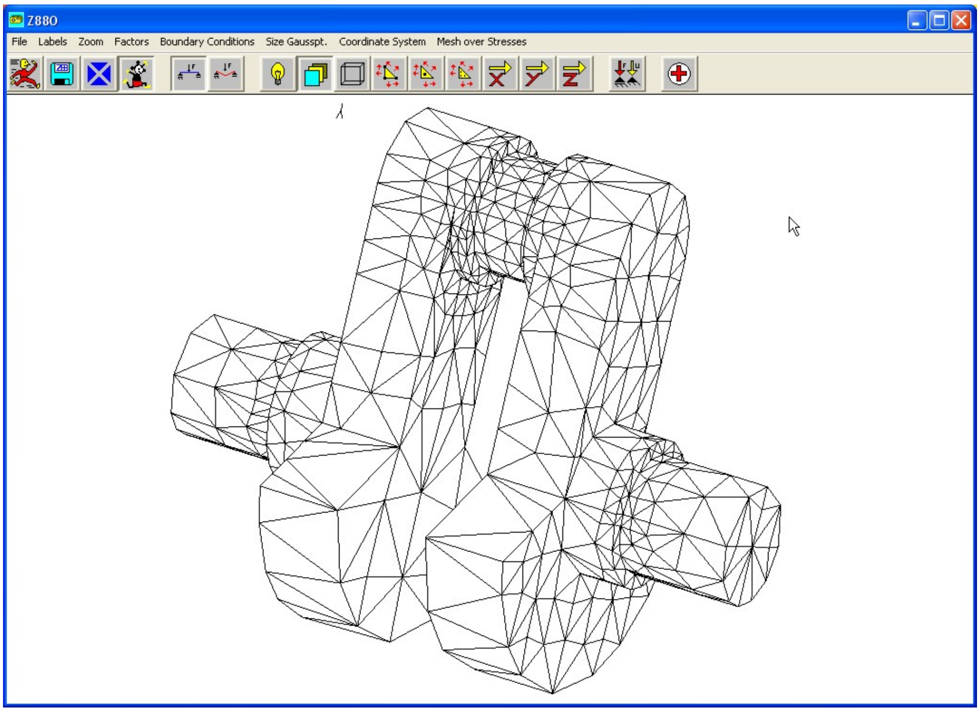
Proceed with a look at the structure with **Z88O**. The computing time with **Z88F** is about 16 sec. on a PC (AMD Athlon 64 X2 3800+ processor, 4 GByte memory, Windows XP). Enter a value of about 11,400,000 for MAXGS.

See the deflected structure with **Z88O**. The angular deflection of the axes 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.

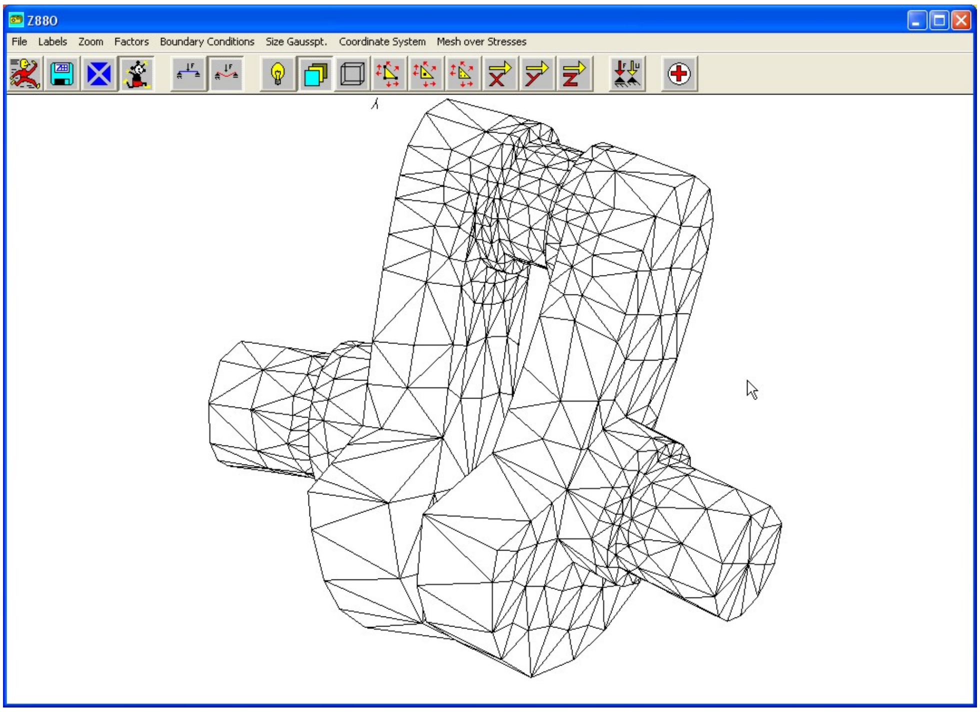


```
Reading Z88I1.TXT :                    Reading material informations
>>> Start Z88A: pass 1 of Z88F <<<     Writing Z88O0.TXT
GS needs 11381064 elements             KOI needs 39420 elements
>>> Start Z88B: pass 2 of Z88F <<<     Compilation no. 3941 type 16
GS needs 11381064 elements             portion zero in GS is 93 %

>>> Start Z88CC: pass 3 of Z88F <<<    Incorporating constraints pass 2
Writing Z88O1.TXT                      Start SCAL88
Start of Cholesky- solver CHOY88
20478 * 20478=size system of equations        20478
Forward-substitution                   Back-substitution
Writing Z88O3.BNY
Writing Z88O2.TXT, Z88F done
```

Windows: Computing deflections with Z88F. Looks quite similar on UNIX machines.



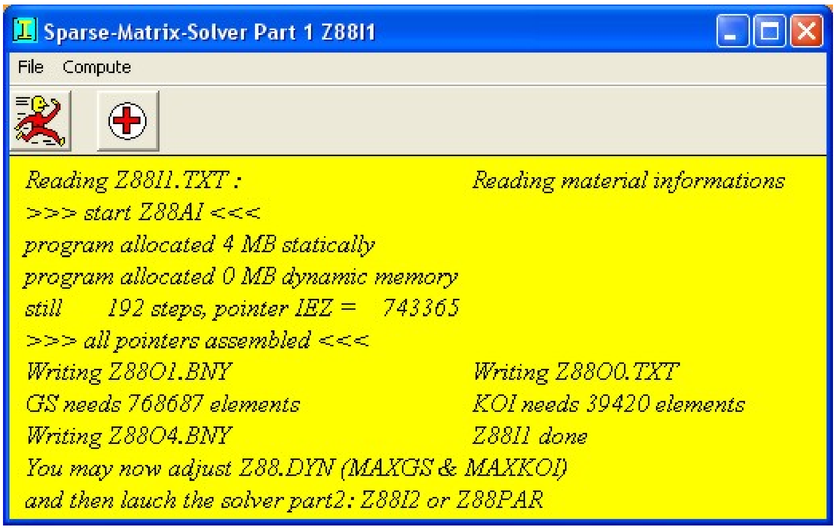
Windows: Plot programm Z88O, undeflected structure.



Windows: Plot programm Z88O, deflected structure.

Now we'll launch the sparse matrix iteration solver Z88I1 and Z88I2. To begin with, we'll try 1,000,000 for MAXIEZ:

```
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
    MAXPR          1     ← has no meaning for this example
    MAXRB        903     ← read off from Z88I2.TXT
    MAXIEZ   1000000     ← important for Z88I1
    MAXGP     500000     ← used by Z88O for the Gauss points
COMMON END
```



```
Reading Z88I1.TXT :                    Reading material informations
>>> start Z88AI <<<
program allocated 4 MB statically
program allocated 0 MB dynamic memory
still     192 steps, pointer IEZ =    743365
>>> all pointers assembled <<<
Writing Z88O1.BNY                      Writing Z88O0.TXT
GS needs 768687 elements               KOI needs 39420 elements
Writing Z88O4.BNY                      Z88I1 done
You may now adjust Z88.DYN (MAXGS & MAXKOI)
and then lauch the solver part2: Z88I2 or Z88PAR
```

Part 1 of the Sparse Matrix Solvers

Our entries did work properly (otherwise, you would have to increase MAXIEZ) and the sorting times was just a breeze. 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.

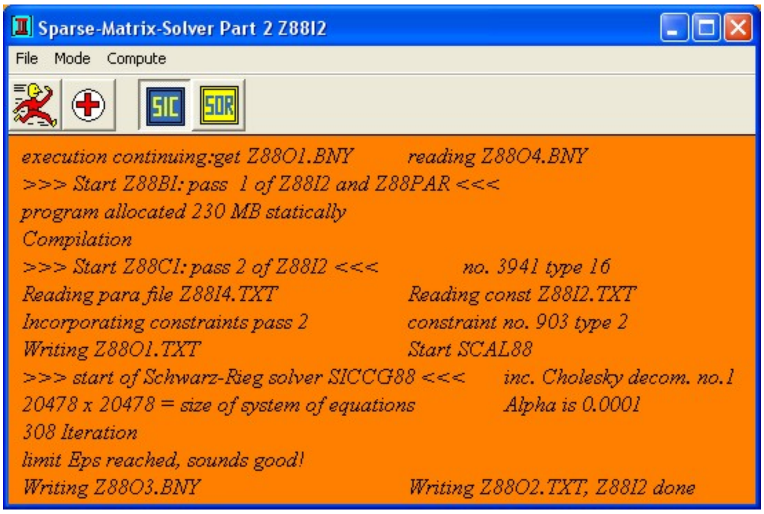
Thus, we would adjust the memory in Z88.DYN as follows (feel free to enter even bigger values):

```
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
    MAXPR          1     ← has no meaning for this example
    MAXRB        903     ← read off from Z88I2.TXT
    MAXIEZ  1000000      ← not used by Z88I2
    MAXGP     500000     ← used by Z88O for the Gauss points
COMMON END
```

If you adjust the iteration parameters in Z88I4.TXT (chapter 3.6) as follows:

```
     10000   1e-7  0.0001  1.  1
```

i.e. a maximum of 10,000 iterations, *EPS* with 1E-7 and α with 0.0001, then this results in a computing time of about 12 sec. on a PC (AMD Athlon 64 X2 3800+ processor, 4 GByte memory, Windows XP). 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 296 iterations and 11 seconds) or 1E-10 (resulting in 329 iterations and 13 sec.), and see the different computing times.

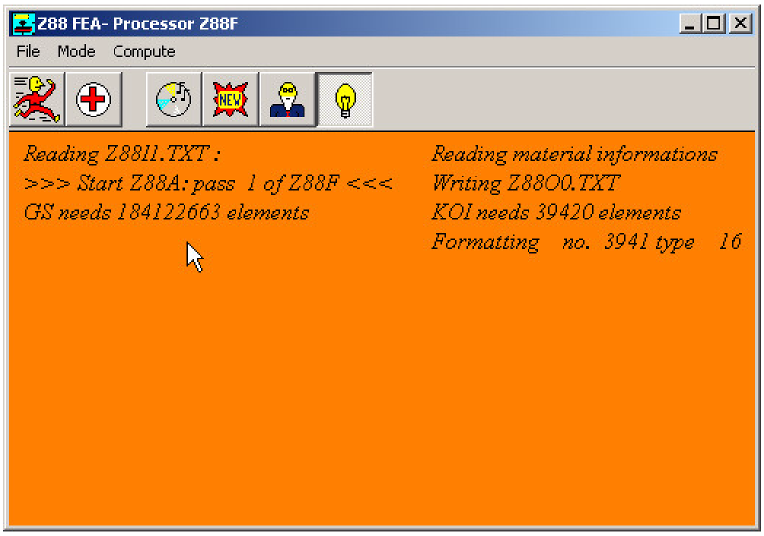


```
execution continuing:get Z88O1.BNY        reading Z88O4.BNY
>>> Start Z88BI: pass  1 of Z88I2 and Z88PAR <<<
program allocated 230 MB statically
Compilation
>>> Start Z88CI: pass 2 of Z88I2 <<<            no. 3941 type 16
Reading para file Z88I4.TXT                Reading const Z88I2.TXT
Incorporating constraints pass 2           constraint no. 903 type 2
Writing Z88O1.TXT                          Start SCAL88
>>> start of Schwarz-Rieg solver SICCG88 <<<     inc. Cholesky decom. no.1
20478 x 20478 = size of system of equations      Alpha is 0.0001
308 Iteration
limit Eps reached, sounds good!
Writing Z88O3.BNY                          Writing Z88O2.TXT, Z88I2 done
```

Windows: The Sparse Matrix Iteration Solver Part 2, i.e. Z88I2.

<u>However, a very nice experiment is this:</u>

Start from the very beginning, run Z88G, but <u>not</u> the Cuthill-McKee algorithm Z88H. Launch directly after Z88G a test run with Z88F:



```
Reading Z88I1.TXT :                    Reading material informations
>>> Start Z88A: pass  1 of Z88F <<<    Writing Z88O0.TXT
GS needs 184122663 elements            KOI needs 39420 elements
                                       Formatting   no.  3941 type   16
```

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!

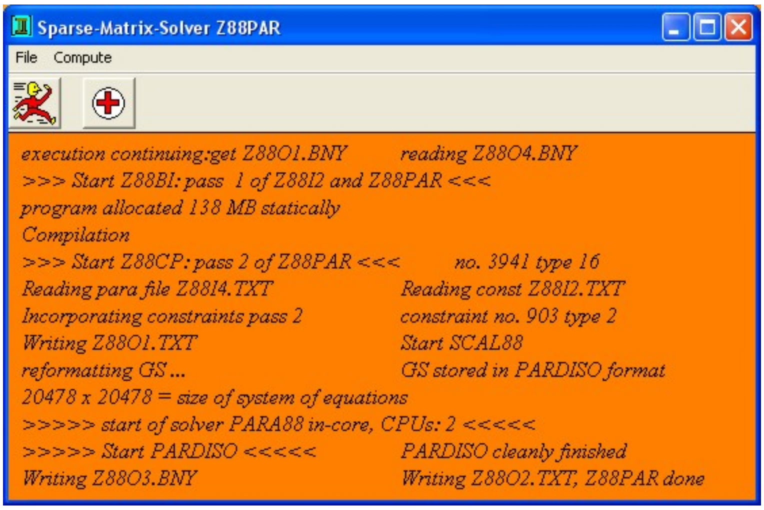
However, run again the iteration solver part 1, i.e. Z88I1. This will again result in only 768,687 elements for the total stiffness matrix. Calculate, please:

184,122,663 : 768,687 = 240 : 1

The second part of the iteration solver, i.e. Z88I2, needs now some more iterations (350 in contrary to 308 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.

Try now the direct Sparse Matrix Solver with Fill-In:

Much faster runs the solver pair Z88I1 and Z88PAR. Adjust the 5th entry in **Z88I4.TXT** to the number of CPUs. Keep in mind that Z88PAR deals heavily with dynamic memory when running. This may cause serious trouble when computing very large structures. For this example the elapsed time was ~ 4 sec. with two CPUs.



```
execution continuing:get Z88O1.BNY        reading Z88O4.BNY
>>> Start Z88BI: pass  1 of Z88I2 and Z88PAR <<<
program allocated 138 MB statically
Compilation
>>> Start Z88CP: pass 2 of Z88PAR <<<        no. 3941 type 16
Reading para file Z88I4.TXT               Reading const Z88I2.TXT
Incorporating constraints pass 2          constraint no. 903 type 2
Writing Z88O1.TXT                         Start SCAL88
reformatting GS ...                       GS stored in PARDISO format
20478 x 20478 = size of system of equations
>>>>> start of solver PARA88 in-core, CPUs: 2 <<<<<
>>>>> Start PARDISO <<<<<                 PARDISO cleanly finished
Writing Z88O3.BNY                         Writing Z88O2.TXT, Z88PAR done
```

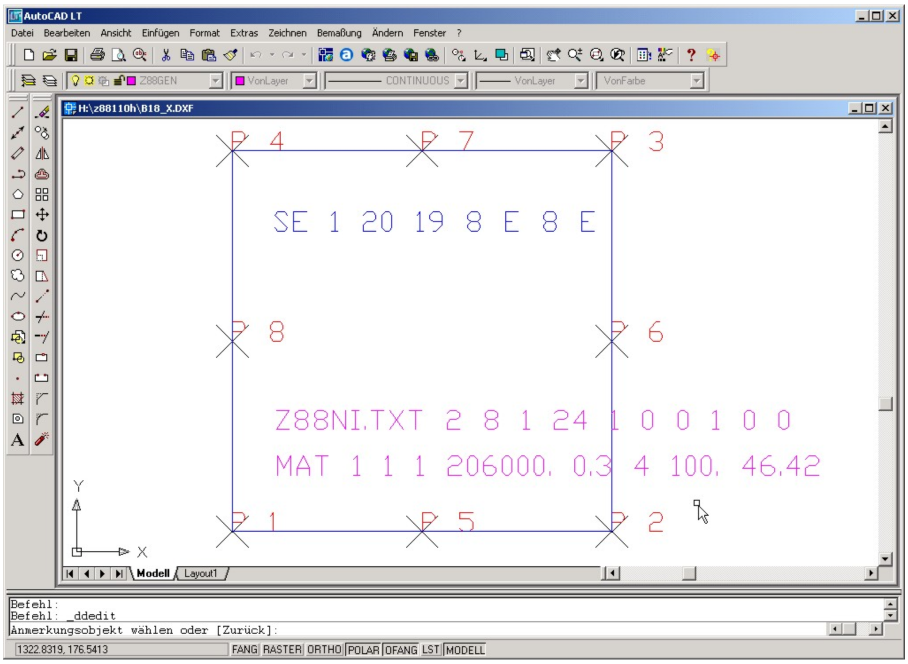
The direct Sparse Matrix Solver Z88PAR.

5.9 RECTANGULAR PLATE, PLATE NO. 19

We want to compute a thick rectangular plate of steel. Data:

- Dimensions: 1,000 × 1,000 × 100 mm
- Surface 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 × 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:



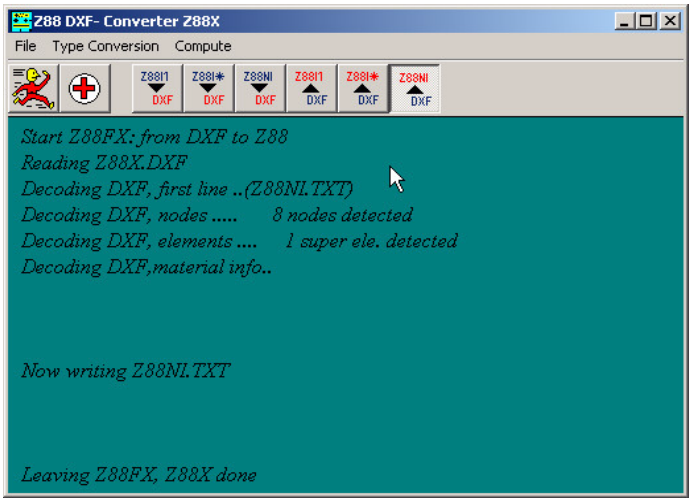
```
SE 1 20 19 8 E 8 E
Z88NI.TXT 2 8 1 24 1 0 0 1 0 0
MAT 1 1 1 206000. 0.3 4 100. 46.42
```

```
Befehl:
Befehl: _ddedit
Anmerkungsobjekt wählen oder [Zurück]:
1322.8319, 176.5413
```

Windows: AutoCAD LT 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):

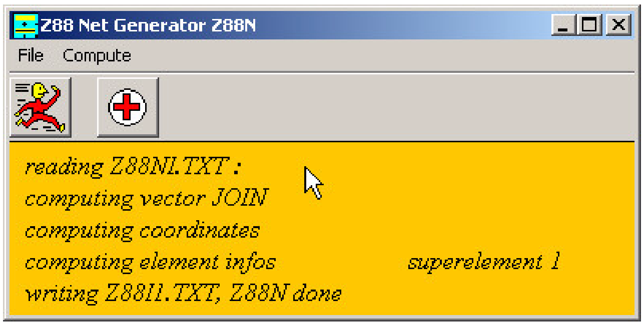
Copy B18_X.DXF into Z88X.DXF



```
Start Z88FX: from DXF to Z88
Reading Z88X.DXF
Decoding DXF, first line ..(Z88NI.TXT)
Decoding DXF, nodes .....       8 nodes detected
Decoding DXF, elements ....     1 super ele. detected
Decoding DXF,material info..


Now writing Z88NI.TXT


Leaving Z88FX, Z88X done
```

Windows: CAD converter Z88X. Looks very similar on UNIX machines.

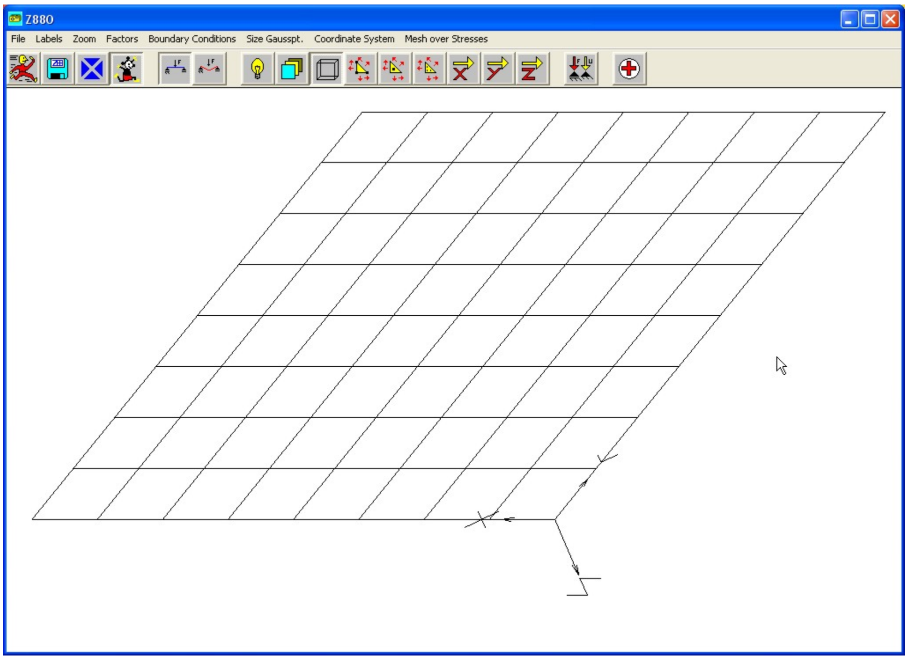
Choose *from Z88X to Z88NI.TXT*. Then, launch the mesher Z88N:



```
reading Z88NI.TXT :
computing vector JOIN
computing coordinates
computing element infos          superelement 1
writing Z88I1.TXT, Z88N done
```

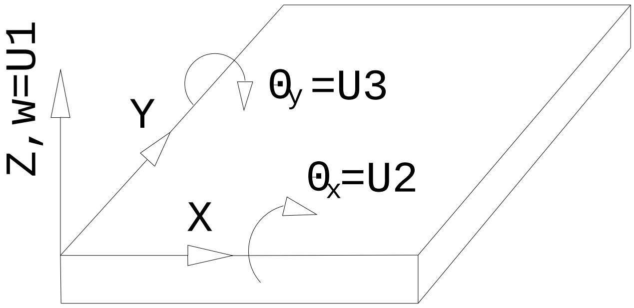
Windows: mesh generator Z88N. Looks very similar on UNIX machines.

Now you may look at the structure with **Z88O**:



Windows: Plot program Z88O, undeflected structure. Looks similar on UNIX computers.

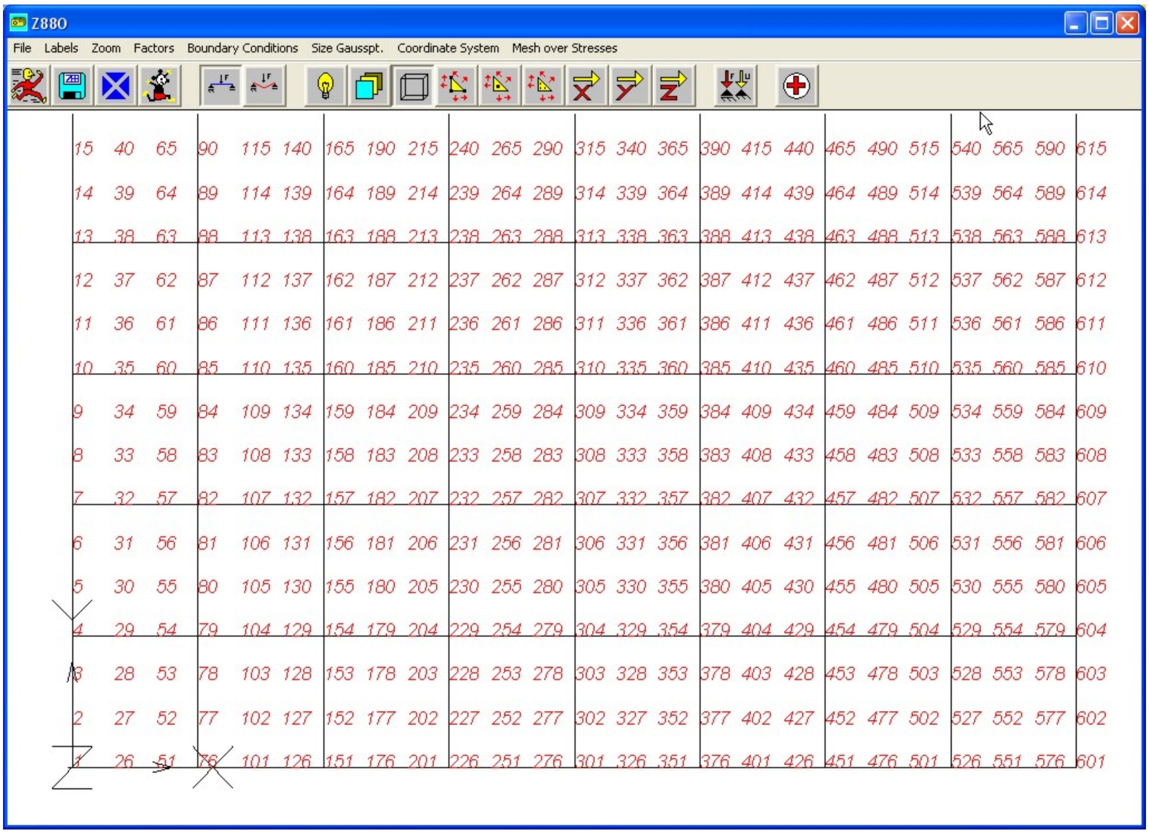
Now you've got some work: you must read off the node numbers for the boundary conditions in Z88O. We have to decide how to support the plate. We'll choose "cutting edges", i.e. the boundaries are supported by a "bezel" above and below. This allows angular movement crosswise to the bezels, but fixture 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



15	40	65	90	115	140	165	190	215	240	265	290	315	340	365	390	415	440	465	490	515	540	565	590	615
14	39	64	89	114	139	164	189	214	239	264	289	314	339	364	389	414	439	464	489	514	539	564	589	614
13	38	63	88	113	138	163	188	213	238	263	288	313	338	363	388	413	438	463	488	513	538	563	588	613
12	37	62	87	112	137	162	187	212	237	262	287	312	337	362	387	412	437	462	487	512	537	562	587	612
11	36	61	86	111	136	161	186	211	236	261	286	311	336	361	386	411	436	461	486	511	536	561	586	611
10	35	60	85	110	135	160	185	210	235	260	285	310	335	360	385	410	435	460	485	510	535	560	585	610
9	34	59	84	109	134	159	184	209	234	259	284	309	334	359	384	409	434	459	484	509	534	559	584	609
8	33	58	83	108	133	158	183	208	233	258	283	308	333	358	383	408	433	458	483	508	533	558	583	608
7	32	57	82	107	132	157	182	207	232	257	282	307	332	357	382	407	432	457	482	507	532	557	582	607
6	31	56	81	106	131	156	181	206	231	256	281	306	331	356	381	406	431	456	481	506	531	556	581	606
5	30	55	80	105	130	155	180	205	230	255	280	305	330	355	380	405	430	455	480	505	530	555	580	605
4	29	54	79	104	129	154	179	204	229	254	279	304	329	354	379	404	429	454	479	504	529	554	579	604
3	28	53	78	103	128	153	178	203	228	253	278	303	328	353	378	403	428	453	478	503	528	553	578	603
2	27	52	77	102	127	152	177	202	227	252	277	302	327	352	377	402	427	452	477	502	527	552	577	602
1	26	51	76	101	126	151	176	201	226	251	276	301	326	351	376	401	426	451	476	501	526	551	576	601

Windows: read off the the nodes with Z88O. 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:

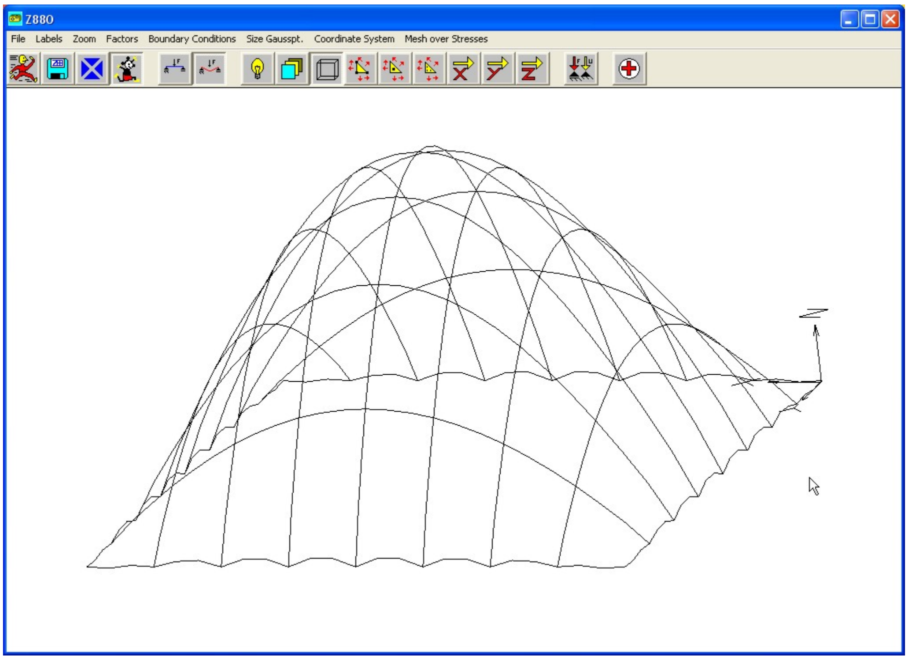
```
    313   +1.1236511E+001   -2.1751298E-008   +2.1751298E-008
```

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:

$$f = \frac{0{,}71\, p\, b^4}{E\, h^3} = \frac{0{,}71 \cdot 46{,}42 \cdot 500^4}{206.000 \cdot 100^3} = 10 \text{ mm}$$

This results in a variety of $\frac{10 - 11{,}24}{10} \cdot 100 = -12\,\%$.

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 <u>all</u> 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!)

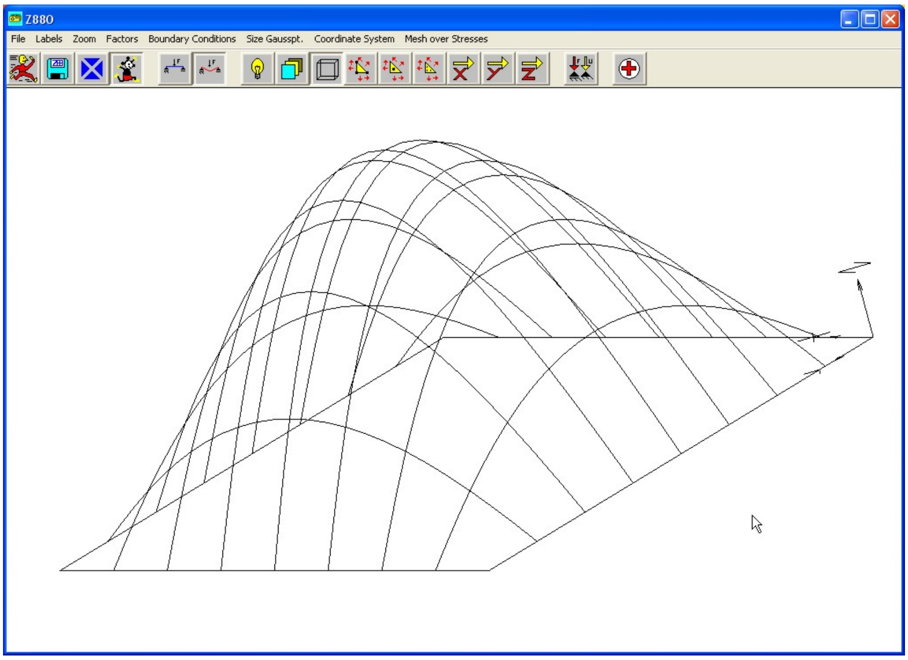
We may calculate the stresses "analytically":

$$\sigma_x = \sigma_y = \frac{1{,}15\, p\, b^2}{h^2} = \frac{1{,}15 \cdot 46{,}42 \cdot 500^2}{100^2} = 1.335 \text{ N/mm}^2$$

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.

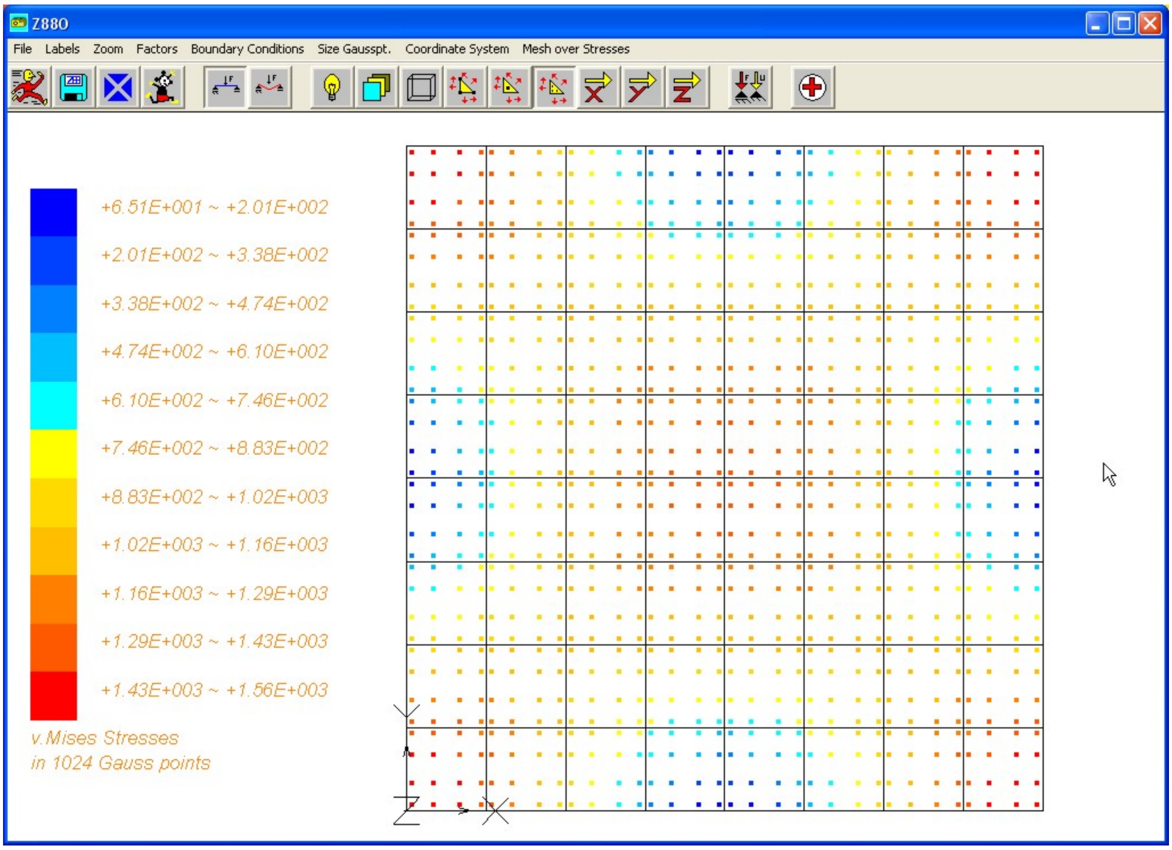


Now the boundaries are supported properly.

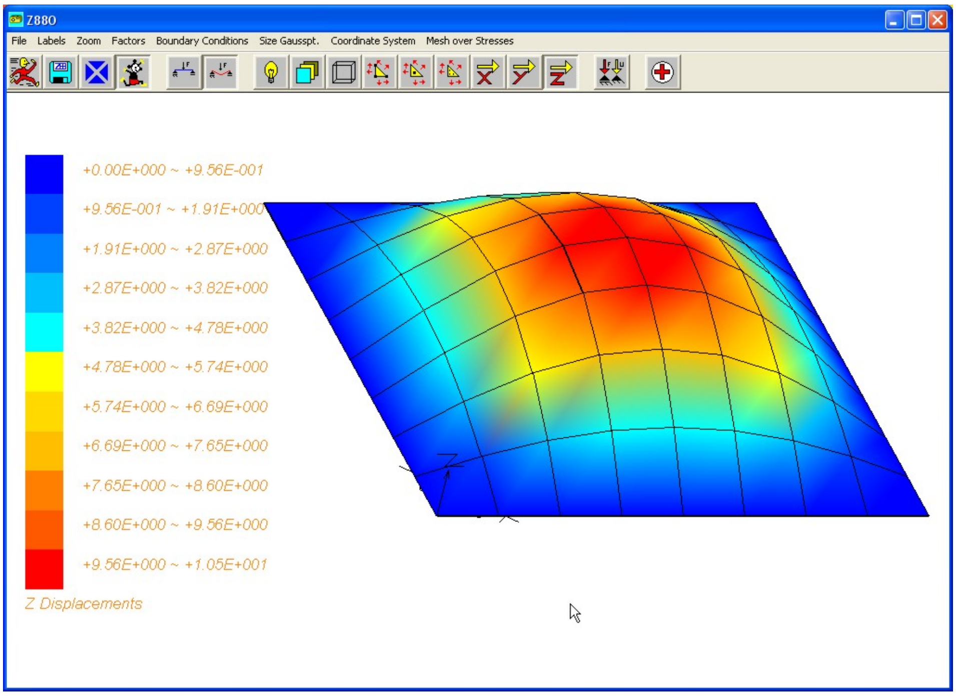
Finally, we'll compute the stresses in the Gauss points and, thus, adjust Z88I3.TXT as follows:

```
    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. Z88O.

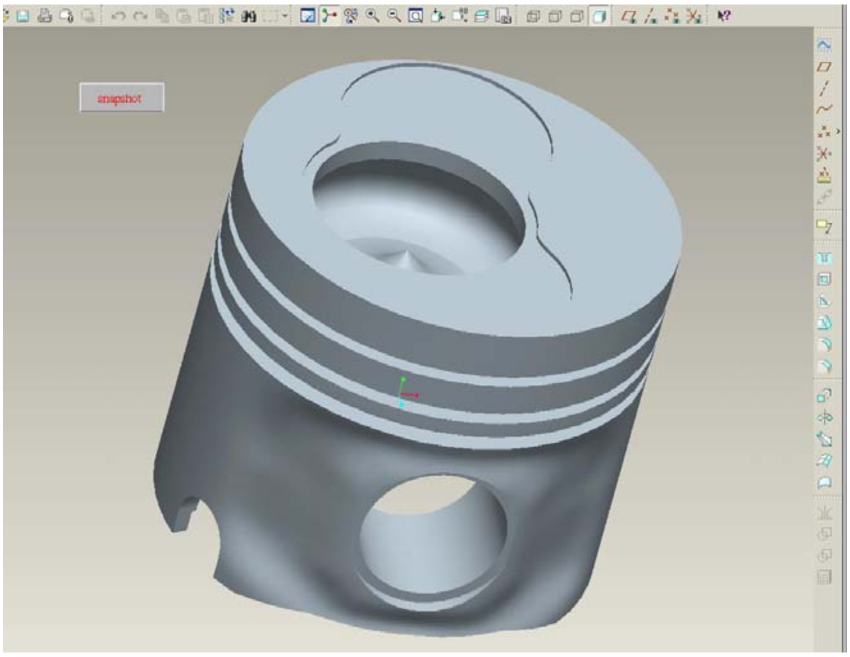


Windows: Plot of the Z displacements. Z88O. Looks similar on UNIX machines.

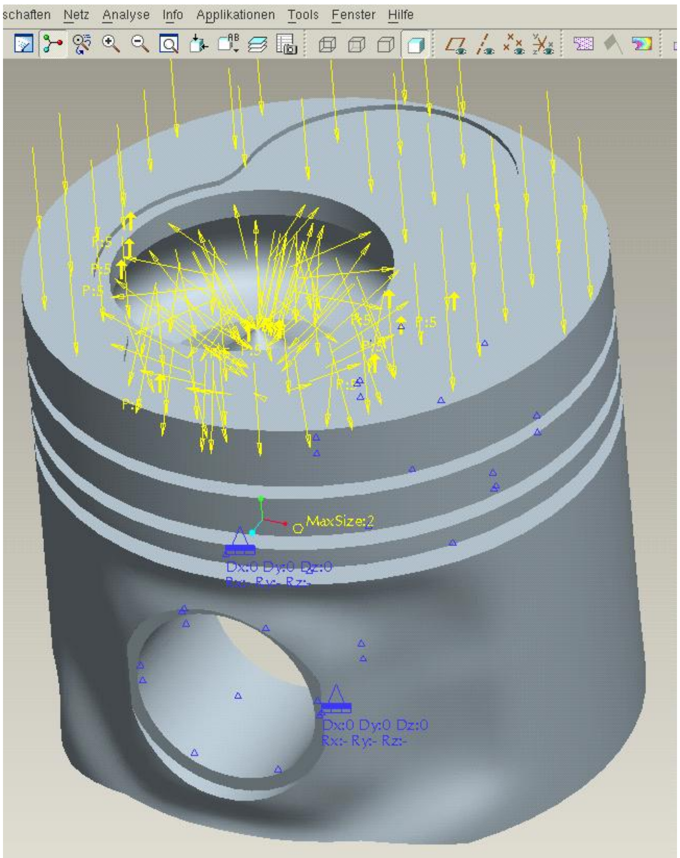
Now you've got a small impression of plate calculation. Consult the devil (and Daniel Webster) when computing deflections and stresses for plates! I recommend parabolic tetrahedrons or hexahedrons in contrary for (thick) plate calculations, that means more input expense but the results are always save and free of suspicious interpretation constraints.

5.10 DIESEL ENGINE PISTON, TETRAHEDRONS NO.16 & 17

This example compares linear shape functions tetrahedrons with 4 nodes and square shape functions tetrahedrons with 10 nodes. However, the pressure load is applied by the surface and pressure loads file Z88I5.TXT. Both the NASTRAN files were compiled with Pro/ENGINEER Wildfire 2:

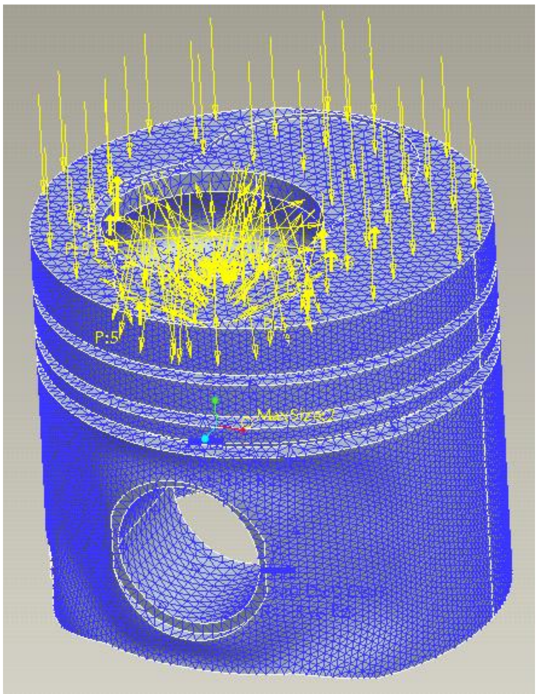


Diesel engine piston of an AUDI engine (simplyfied), modelled by Dipl.-Ing. Jens-Uwe Goering.



Diesel engine piston with pressure load of 50 bar, max. mesh size 2mm.

The piston was modelled similar to the pistons of modern AUDI diesel engines. The pressure load of 50 bar = 5 N/mm^2 and the light alloy material with $E = 73{,}000$ N/mm^2 und $\nu = 0.33$ were chosen with arbitrariness. Of course, in reality higher pressures and other kinds of light alloy are used – but this is not important for our test runs here. We compiled a fine-meshed structure by allowing a max. mesh size of only 2 mm in Pro/ENGINEER.



The compiled mesh resulting in ~ 280,000 tetrahedrons.

Here we go with linear shape functions tetrahedrons. For your convenience a NASTRAN input file *B21_LIN_G.NAS* is prepared and Z88.DYN should look as follows:

```
  COMMON START
    MAXGS    3600000
    MAXKOI   1120000
    MAXK       58000
    MAXE      280000
    MAXNFG    172000
    MAXNEG        32
    MAXPR      50000
    MAXRBD      4000
    MAXIEZ   3600000
    MAXGP    1200000
  COMMON END
```

The surface and pressure loads file Z88I5.TXT looks as follows (please check with the chapters 3.7 and 4.17):

```
4430   Z88I5.TXT,via Z88G V13 NASTRAN
  265 +5.00000E+000   731   728   732
  292 +5.00000E+000   344   345   847
  525 +5.00000E+000 16105 16106 15009
  640 +5.00000E+000 15582 15584 15583
  658 +5.00000E+000 15582 15548 15547
  701 +5.00000E+000   812   817   815
  .........
```

Part 1 of the sparse matrix solver Z88I1 needs 31 MB memory, part 2 of the sparse matrix solver Z88I2 needs 89 MB if you'll choose the Cholesky preconditioning with an $\alpha = 0.0001$. Then, the solver does 202 iterations and will finish the job on a modern PC running Windows XP within one minute.

Z88 computes: $\sigma_{vonMises} = 35.1$ N/mm2 $y_{max} = -0.0121$ mm

Now we'll run the job with square shape functions tetrahedrons resulting in this Z88.DYN:

```
  COMMON START
    MAXGS   51000000
    MAXKOI   2800000
    MAXK      416000
    MAXE      280000
    MAXNFG   1250000
    MAXNEG        32
    MAXPR      50000
    MAXRBD     12000
    MAXIEZ  51000000
    MAXGP    1500000
  COMMON END
```

Use the NASTRAN input file *B21_PARA_G.NAS.*

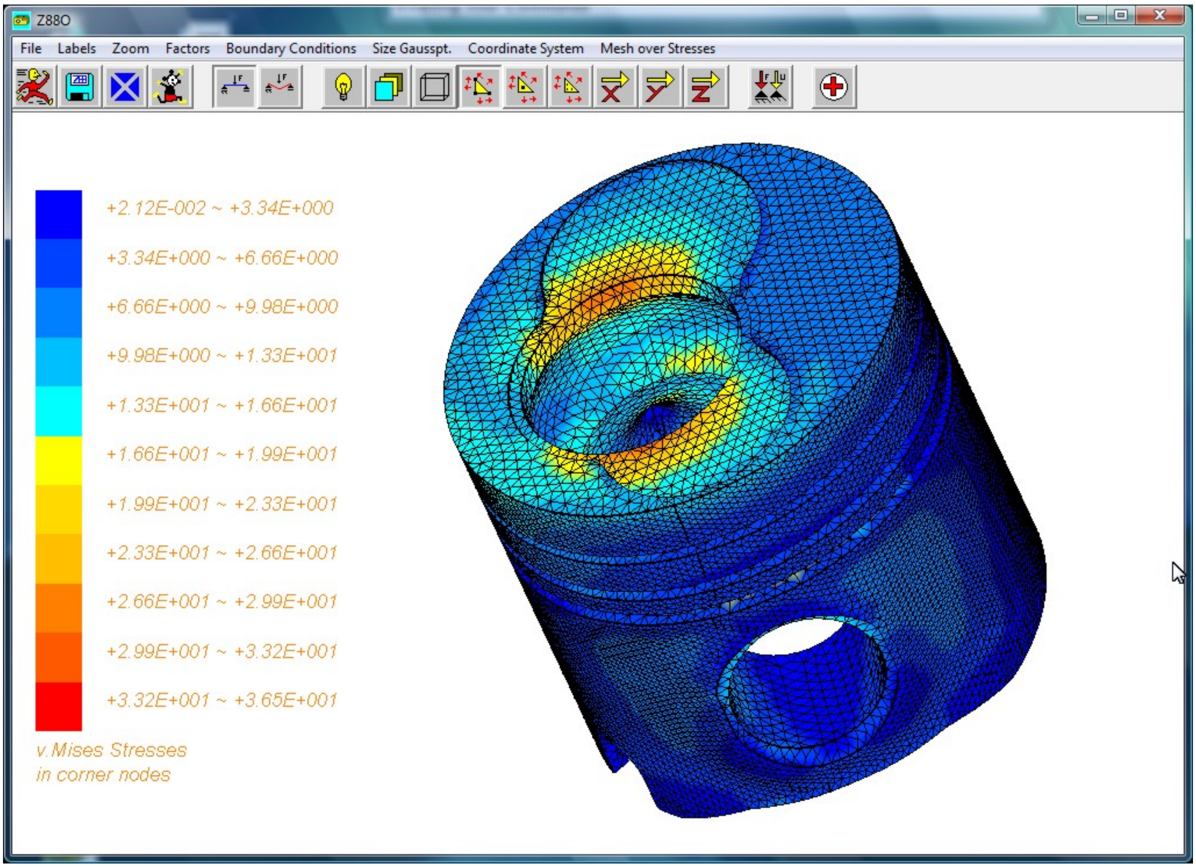
The surface and pressure loads file Z88I5.TXT looks as follows (please check with the chapters 3.7 and 4.16):

```
4430   Z88I5.TXT,via Z88G V13 NASTRAN
    5 +5.00000E+000   394   734   610 59815 61330 59813
```

```
128 +5.00000E+000 16135 16138 16136 167350 167355 167348
292 +5.00000E+000 15401 15400 15399 162081 162074 162075
369 +5.00000E+000 15319 15302 15317 161397 161396 161503
379 +5.00000E+000   828   833   831 63009 63029 63008
682 +5.00000E+000 15582 15548 15547 163056 163041 163044
.........
```

Part 1 of the sparse matrix solver Z88I1 needs 250 MB memory, part 2 of the sparse matrix solver Z88I2 needs 1,070 MB if you'll choose the Cholesky preconditioning with an $\alpha = 0.0001$ (you may reduce this amount by ~1/3 if you'll choose the SOR preconditioning with an $\omega = 1.2$). Then the solver does 668 iterations and finishes the run on a PC with an AMD Athlon 64 X2 3800+ and 4 GByte memory running Windows XP in 45 min.

Z88 computes: $\sigma_{vonMises} = 36.5$ N/mm2 $y_{max} = -0.0128$ mm



Stresses plotted by Z88O for tetrahedrons No.16.

As you see the results differ only minimally and the big time and memory expense for the square shape functions tetrahedrons No.16 was completely useless. But just this is the art of finite elements computing – to choose the best suitable element types!