HANSER

Sample Pages

Frank Rieg, Reinhard Hackenschmidt, Bettina Alber-Laukant

Finite Element Analysis for Engineers

Basics and Practical Applications with Z88Aurora

ISBN (Buch): 978-1-56990-487-9

ISBN (E-Book): 978-1-56990-488-6

Preface

Following the ongoing strong demand in the last years for an English version of the German standard work “Finite Elemente Analyse für Ingenieure” we decided to satisfy this.

Our aim with this book is:

To provide well-chosen aspects of the finite elements for a student of engineering sciences from the 3rd semester and an engineer already established in the job in such a way that he can apply this knowledge immediately to the solution of practical problems.

Therefore, already in the title of the book we speak of finite element analysis (FEA) and not of finite element method. This gigantic field has left behind the quite dubious air of a method for a long time and today is *the* engineer’s tool to analyse structures. Of course, one can do much more with this process than mechanics: heat flows, electric fields and magnetic fields, actually, differential equations and boundary problems for different fields in general – all of this can be solved with it.

However, everything has begun with the calculation of mechanical structures and, hence, we want to limit ourselves in this book to linear and non-linear statics, stationary heat conduction and natural frequencies. The engineer’s aspect is very substantial to us – it does not appear in the title of this book without any reason: The process was developed fairly “intuitively” in the fifties by airplane engineers for static calculations of airplane structures. It is a process from engineers for engineers!

Hence, we proceed as follows: After a really easy demonstration of the basic procedure, we will discuss the most important points of the elasticity theory, the engineering mechanics and the thermodynamics, as far as the FEA is concerned. With this knowledge we continue with the derivation of the element stiffness matrices. This theoretical knowledge is indispensable for proper and clever working with FEA programs. Then we look at the compilation procedure, at the storage processes and at the solving of the equation systems to calculate the unknowns.

In order to transfer your knowledge into practice, we have put two FE programs on DVD: Z88®, the open source finite elements program for static calculations, programmed by the lead author of this book, as well as Z88Aurora®, the very comfortable to use and much more powerful freeware finite elements program which can also be used for non-linear calculations, stationary heat flows and natural frequencies. Both are full versions with which *arbitrarily big structures* can be computed. The only limits are given by your computer concerning main storage and disc storage and by your powers of imagination. Z88 and Z88Aurora are ready-to-run for Windows, LINUX, as well as for Mac OS X. For Z88 we directly provide the sources, so that you can study the theoretical aspects in the program code and extend it if necessary. This way, you can also understand the working of memory processes, equations solvers and so forth. Z88 is transparent for the user through input and output via text files. It is a FEA program in the quite classical and original sense. In addition, we think: You only learn the basics with a program like this, as every numerical value can and has to be controlled. As soon as you have understood the basic procedure, you can work with Z88Aurora, which was developed at our *Chair of Engineering Design and CAD* at the University of Bayreuth, Germany, with promotion of the *Oberfrankenstiftung*. Z88Aurora does not take second place in *look and feel* compared to the commercial FEA programs and allows a very professional and contemporary work, directly from CAD data. We do not refer to the known commercial FEA programs here because the versions that are free of charge only offer very limited options concerning the structure sizes with which you could not compute several of the following examples at all. Moreover, we cannot offer source codes for them. In later sections of the book there are many practical examples that we recommend to check. The DVD also contains the input files for all examples. The examples are selected in a way that gradually explains the different aspects of the calculation of structures and mechanical structures.

Furthermore, we have developed an app for Android devices called Z88Tina (*www.z88tina.de*) which is a very, very small cousin of our full-featured freeware FEA program Z88Aurora (*www.z88.de*) and is derived from the open source FEA program Z88V14OS. Z88Tina can be downloaded from Google Play Store: *https://play.google.com/store/apps/details?id=z88tina.fr*

For this fourth German edition (and first English edition) we have completely revised our book on finite element analysis: The theoretical section has been extended concerning shell elements (by Prof. F. Rieg, PhD), non-linear calculations (by C. Wehmann, PhD), stationary heat conduction (by M. Frisch, M.Sc.) and natural frequencies (by M. Neidnicht, PhD). The examples have been strongly extended and updated. Our employees M. Frisch, M.Sc., M. Neidnicht, PhD, F. Nützel, M.Sc., C. Wehmann, PhD, J. Zapf, PhD, and M. Zimmermann, M.Sc., did the programming and testing of Z88Aurora version 2 and gave valuable recommendations for the text of this book. We wish to thank them all a lot. Our very special thanks is directed towards Kevin Deese and Christoph Wehmann for their systematic translation error search. It was a hell of a work. We also thank our publishing house Carl Hanser Verlag for the exemplary realization of this book.

The work on this book was again a pleasure to us and we hope you will enjoy this book.

Frank Rieg, Reinhard Hackenschmidt and Bettina Alber-Laukant

Bayreuth, Germany, June 2014

Contents

Preface v

1 **Introduction** 1

2 **The Basic Procedure** 5

3 **Some Elasticity Theory** 23

3.1 Displacements and Strains 23

3.1.1 For the Truss 23

3.1.2 For Plane Stress 25

3.1.3 In Space 31

3.1.4 For the Plate 32

3.2 Stress-Strain Relations 34

3.3 Basics of Thermomechanical Loading 44

3.4 Basic Principles of Natural Vibration 47

3.5 Basic Principles of Non-linear Calculations 50

4 **Finite Elements and Element Matrices** 63

4.1 Basics of Element Stiffness Matrices 65

4.2 Constitutive Matrices 69

4.3 B Matrix 70

4.4 Shape Functions 71

4.5 Integration 81

4.6 The Application of Loads, Load Vectors 88

4.6.1 The Basic Procedure 88

4.6.2 Plate Elements 91

4.6.3 Volume Elements 93

4.6.4 Plane and Axial-Symmetrical State of Stress 104

4.6.5 Distributed Loads for Beams 106

4.6.6 Gerber Joints for Beams 108

4.7 A complete Element Stiffness Routine 112

4.8 Some Remarks on Modelling 121
4.8.1 Choice of Element Types 121
4.8.2 Polymers and Material Laws 129
4.8.3 Structural Optimization 130
4.9 Some Remarks on Shells 134
4.10 Element Matrices for Heat Transfer 148
4.11 Element Matrices for Vibration 150
4.12 Element Matrices of the Non-linear Finite Element Analysis 152

5 Compilation, Storage Schemes and Boundary Conditions 163
5.1 Compilation 163
5.2 Storage Schemes 174
5.2.1 Band Width Storage Scheme 176
5.2.2 The Skyline Storage Scheme 180
5.2.3 The Jennings Storage Scheme 182
5.2.4 The Non-Zero Storage Scheme 190
5.2.5 Summary of the Storage Schemes 196
5.3 Boundary Conditions 197
5.3.1 Single Forces and Single Displacements 197
5.3.2 Distributed Loads with Plates 200
5.3.3 Fixture of plates 202
5.3.4 Boundary Conditions in Temperature Analyses 203
5.3.5 Boundary Conditions with Vibration 206
5.3.6 Boundary Conditions in the Non-linear Finite Element Analysis 207

6 Solvers 209
6.1 Direct Solvers 210
6.1.1 The Cholesky Solver 212
6.2 Condition and Scaling 214
6.3 Iterative Solvers 223
6.3.1 The Jacobi Method 225
6.3.2 The Gauss-Seidel Method 226
6.3.3 The SOR Method and the JOR Method 226
6.3.4 The basic CG Solver 227
6.3.5 The CG Solver with Pre-conditioning 229
6.4 Solver for Thermomechanical Problems 244
6.5 Solver for Vibration Problems 244
6.6 Solver for the Non-linear Finite Element Analysis 254

7 Stresses and Nodal Forces 257
7.1 Stresses 257
7.2 Reduced Stresses 264
7.3 Nodal Forces 271

8 Mesh Generation of Curvilinear Finite Elements 275
8.1 Basis Considerations of the Procedure 275
8.2 Mathematical Foundations 277
8.3 Description of a Simple Mapped Mesher 281

9 Z88: The Basics 289
9.1 General Information 289
9.1.1 Summary of the Z88 Element Library 290
9.2 The Open Source FE Program Z88 302
9.2.1 Overview of the Z88 Program Modules 302
9.2.2 Dynamic Memory Z88 305
9.2.3 The Input and Output of Z88: 308
9.3 The Freeware FE Program Z88Aurora 312
9.3.1 Overview of the Z88Aurora Modules 312
9.3.2 Memory Requirement in Z88Aurora 315
9.3.3 The Input and Output of Z88Aurora 316

10 Z88: The Modules 319
10.1 The Linear Solver Z88R 319
10.1.1 Z88R: The Cholesky Solver 320
10.1.2 Z88R: The Sparse Matrix Solvers SICCG and SORCG 321
10.1.3 Z88R: The Sparse Matrix multi-core Solver PARDISO 323
10.1.4 Which Solver to choose? 324
10.1.5 Explanations for Stress Calculations 324
10.1.6 Explanations for Nodal Force Calculations 325
10.2 The Mapped Mesher Z88N 325
10.3 The Advanced Mapped Mesher in Z88Aurora 328
10.3.1 The Use of Z88N in Z88Aurora 328
10.3.2 Tetrahedron Refiner Z88MTV 329
10.3.3 The 2D Shell Thickener Z88MVS 331
10.4 The OpenGL Plot Program Z88O in Z88 V14 OS or the Post-Processor of Z88Aurora 331
10.5 The DXF Converter Z88X 335
10.6 The 3D Converter Z88G 344
10.7 The Ansys Converter Z88ASY in Z88Aurora 347
10.8 The Abaqus Converter Z88INP in Z88Aurora 349
10.9 Das Cuthill-McKee Program Z88H 350
10.10 The STEP Import Z88GEOCON (STEP) in Z88Aurora 352
10.11 The STL Converter Z88GEOCON (STL) in Z88Aurora 354
10.12 The Tetrahedron Mesher in Z88Aurora 355
10.13 The Picking Module of Z88Aurora 356
10.14 The Material Data Base of Z88Aurora 358
10.15 Applying Boundary Conditions in Z88Aurora 358

10.16 The User Support with Spider in Z88Aurora 359
10.17 The Thermomechanical Solver in Z88Aurora 360
10.18 The free Vibration Solver in Z88Aurora 363
10.19 The Non-linear Solver Z88NL of Z88Aurora 366

11 Generating Input Files 371
11.1 General Information 371
11.2 General Structure Data File Z88I1.TXT 373
11.3 Boundary Condition File Z88I2.TXT 374
11.4 Surface and Pressure Loads File Z88I5.TXT 377
11.5 Material Parameters File Z88MAT.TXT 382
11.6 Material Data File *.TXT 383
11.7 Element Parameters File Z88ELP.TXT 383
11.8 Integration Order File Z88INT.TXT 385
11.9 Mapped Mesher Input File Z88NI.TXT 386
11.10 Solver Parameters File Z88MAN.TXT 390
11.11 Comparison of the different Z88 Data File Formats 393

12 The Finite Elements of Z88 and Z88Aurora 395
12.1 Hexahedron No. 1 with 8 Nodes 395
12.2 Beam No. 2 with 2 Nodes in Space 398
12.3 Plane Stress Element No. 3 with 6 Nodes 400
12.4 Truss No. 4 in Space 401
12.5 Shaft No. 5 with 2 Nodes 402
12.6 Torus No. 6 with 3 Nodes 404
12.7 Plane Stress Element No. 7 with 8 Nodes 405
12.8 Torus No. 8 with 8 Nodes 407
12.9 Truss No. 9 in the Plane 409
12.10 Hexahedron No. 10 with 20 Nodes 411
12.11 Plane Stress Element No. 11 with 12 Nodes 414
12.12 Torus No. 12 with 12 Nodes 416
12.13 Beam No. 13 in the Plane 418
12.14 Plane Stress Element No. 14 with 6 Nodes 419
12.15 Torus No. 15 with 6 Nodes 421
12.16 Tetrahedron No. 16 with 10 Nodes 424
12.17 Tetrahedron No. 17 with 4 Nodes 427
12.18 Plate No. 18 with 6 Nodes 429
12.19 Plate No. 19 with 16 Nodes 431
12.20 Plate No. 20 with 8 Nodes 434
12.21 Shell No. 21 with 16 Nodes 436
12.22 Shell No. 22 with 12 Nodes 438
12.23 Shell No. 23 with 8 Nodes 440

12.24 Shell No. 24 with 6 Nodes 442
12.25 Element/Solver Overview Z88Aurora V2 444

13 Examples 445
13.1 Flat Wrench (Plate No. 7) 452
13.1.1 With Z88 V14 453
13.1.2 With Z88Aurora V2 461
13.2 Crane Girder made of Trusses No. 4 471
13.2.1 With Z88 V14 472
13.2.2 With Z88Aurora V2 477
13.3 Gear Shaft with Shaft No. 5 482
13.3.1 With Z88 V14 484
13.3.2 With Z88Aurora V2 487
13.4 Bending Girder with Beam No. 13 491
13.4.1 With Z88 V14 492
13.4.2 With Z88Aurora V2 496
13.5 Plate Segment of Hexahedrons No. 1 and No. 10 500
13.5.1 With Z88 V14 501
13.5.2 With Z88Aurora V2 507
13.6 Pipe under Internal Pressure, Plain Stress Element No. 7 510
13.6.1 With Z88 V14 511
13.6.2 With Z88Aurora V2 518
13.7 Pipe under Internal Pressure, Torus No. 8 520
13.7.1 With Z88 V14 521
13.7.2 With Z88Aurora V2 527
13.8 Two-Stroke Engine Piston 529
13.8.1 With Z88 V14 530
13.8.2 With Z88Aurora V2 534
13.9 RINGSPANN Spring and Belleville Spring 539
13.9.1 With Z88 V14 541
13.9.2 With Z88Aurora V2 544
13.10 Liquid Gas Tank 546
13.10.1 With Z88 V14 546
13.10.2 With Z88Aurora V2 550
13.11 Motorcycle Crankshaft 552
13.11.1 With Z88 V14 554
13.11.2 With Z88Aurora V2 559
13.12 Torque-measuring hub 563
13.12.1 With Z88 V14 564
13.12.2 With Z88Aurora V2 565
13.13 Plane Frameworks 566
13.13.1 With Z88 V14 567
13.13.2 With Z88Aurora V2 587

13.14 Gearwheel 589
13.14.1 With Z88 V14 590
13.14.2 With Z88AuroraV2 595
13.15 3D Wrench 599
13.15.1 With Z88 V14 599
13.15.2 with Z88Aurora V2 611
13.16 Force Measuring Element, Plane Stress Elements No. 7 613
13.16.1 With Z88 V14 613
13.16.2 With Z88Aurora V2 623
13.17 Circular Plate, Plates No. 20 624
13.17.1 With Z88 V14 626
13.17.2 With Z88Aurora V2 630
13.18 Rectangular Plate with 16 Nodes Plates No. 19 631
13.18.1 With Z88 V14 631
13.18.2 With Z88Aurora V2 638
13.19 Four-stroke Engine Pistons with Tetrahedrons No. 16 639
13.19.1 With Z88 V14 640
13.19.2 With Z88Aurora V2 644
13.20 Motorcar Fan Wheel 647
13.20.1 With Z88 V14 649
13.20.2 With Z88Aurora V2 650
13.21 Diesel Piston 653
13.21.1 With Z88 V14 654
13.21.2 With Z88Aurora V2 656
13.22 Calculation of a Stress Concentration Factor 657
13.22.1 With Z88 V14 658
13.22.2 With Z88Aurora V2 663
13.23 Gear Root Stress 664
13.23.1 With Z88 V14 666
13.23.2 With Z88Aurora V2 668
13.24 Square Pipe, Shell No. 24 670
13.24.1 With Z88 V14 671
13.24.2 With Z88Aurora V2 673
13.25 Submarine made of Shells No. 22 677
13.26 Gear Wheel out of Tetrahedrons No. 17 682
13.27 Oscillating Drum 685
13.28 Modal Analysis Crankshaft 689
13.29 Thermo-mechanical Analysis of a Spoon 692
13.30 Thermal Analysis of a four-stroke Engine Piston 698
13.31 Non-linear Calculation of a Belleville Spring 702
13.32 Non-linear Calculation of a Hinge 706

References and further reading **711**
Index **717**

The DVD that comes with the book *Finite Element Analysis for Engineers* contains the program versions Z88 V14 OS and Z88Aurora V2 including all data necessary to use the examples of both versions. The content of the DVD is organized as follows:

/z88_examples_z88aurora/:	Examples for Z88Aurora V2
/z88_examples_z88v14os/:	Examples for Z88 V14 OS
/z88aurora/:	Installer and documentation Z88Aurora V2
/z88v14os/:	Unzipped directories Z88 V14 OS

Installation of Z88 V14 OS

Z88 V14 OS is available as a ready-to-run version as well as a version for self-compiling in the directory */z88v14os/* for the following operating systems:

- 32 BIT Windows
- 64 BIT Windows
- 32 BIT LINUX
- 64 BIT LINUX
- 64 BIT Mac OS X

In the file *z88mane.pdf* in the directory */z88v14os/docu/* you find the detailed documentation for installation and compiling.

Installation of Z88Aurora V2

Z88Aurora V2 is available in the directory */z88aurora/* as installer for

- 32 BIT Windows and
- 64 BIT Windows

and as TAR.GZ for

- 64 BIT LINUX Suse 12.1 and 12.2
- 64 BIT LINUX Ubuntu 11.04, 12.04 and 14.04
- 64 BIT Mac OS X ex 10.6 (Please note that when using UNIX und Mac the access rights have to be adapted.)

In the directory */z88aurora/installer/* you find the detailed installation manual for the corresponding operation system.

Please note, that when using Mac OS X the GTK+-package gtk+4z88.dmg (which you find in the directory */z88aurora/installer/macosx*) has to be installed at first.

In the directory */z88aurora/docu/* you find the theory manual and the user guide.

Software Updates

The DVD’s software status is June 10th, 2014.

On *www.z88.de* you can find the user forum as well as updates and error corrections.

3 Some Elasticity Theory

3.1 Displacements and Strains

3.1.1 For the Truss

When looking into books on technical mechanics or FEA we often find the following:

$$\varepsilon_x = \frac{\partial u}{\partial x}$$

This is often accompanied by the remark “as one sees immediately”. We never considered such equations as “immediately reasonable”; hence, the derivation of the so-called relation of deformation and displacement is here presented in detail. It is the basis for understanding the continuum elements of FEA. With this, we lean upon the excellent book of *Bickford /10/*, but we also recommend the lecture of *Love /8/*, *Timoshenko /9/* and *Schnell/Gross/Hauger /90/*.

We act on the assumption of a simple rubber band (which of course could also be a steel tape) and pull it with a force F. The origin length of the rubber band is ℓ_0, the stretched tape has the length ℓ_1. The extension of the tape is called $\Delta\ell$.

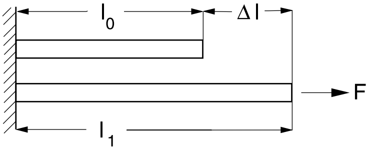

Figure 3.1-1: Length change of a rod by the force effect

We define the strain $\varepsilon = \dfrac{\Delta\ell}{\ell_0}$

with $\Delta\ell = \ell_1 - \ell_0$ and $\varepsilon = \dfrac{\ell_1 - \ell_0}{\ell_0} = \dfrac{\Delta\ell}{\ell_0}$.

To examine the strain in every point, we select two points A and B on the tape, which are located very closely together, and call it the distance Δx.

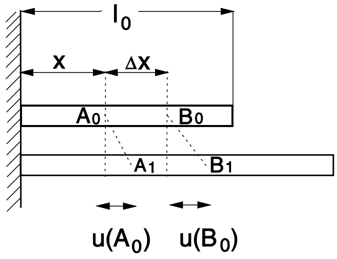

Figure 3.1-2: Selective consideration of the displacements u in A and B

According to the definition $\varepsilon = \frac{\Delta \ell}{\ell} = \frac{\ell_1 - \ell_0}{\ell_0}$.

By implication, the strain in A_0 is

$$\varepsilon_x(A_0) = \lim_{A_0B_0 \to 0} \frac{A_1B_1 - A_0B_0}{A_0B_0}$$

with A_1B_1 representing the distance between A_1 and B_1 resp. A_0B_0 representing the distance between A_0 and B_0.

With $A_0B_0 = \Delta x$ and

$$A_1B_1 = (x + \Delta x + u(B_0)) - (x + u(A_0)) = \Delta x + u(B_0) - u(A_0)$$

is

$$\varepsilon_x(A_0) = \lim_{A_0B_0 \to 0} \frac{A_1B_1 - A_0B_0}{A_0B_0} = \lim_{A_0B_0 \to 0} \frac{\Delta x + u(\mathrm{B}_0) - u(A_0) - \Delta x}{\Delta x}$$

We can call the difference $u(B_0) - u(A_0)$ Δu and get:

$$\varepsilon_x(A_0) = \lim_{A_0B_0 \to 0} \frac{\Delta x + \Delta u + \Delta x}{\Delta x} = \lim_{A_0B_0 \to 0} \frac{\Delta u}{\Delta x} \lim_{\Delta x \to 0} \frac{\Delta u}{\Delta x}$$

and in the limiting process:

$$\varepsilon_x(\mathrm{A}_0) = \frac{du}{dx} \quad \text{in } A_0$$

or in general:

$\varepsilon = u'$ “Strain-deflection function”

This means: The strain (or expansion) ε is the derivation of the deflection function $u(x)$. Thus

$$\varepsilon_x = u' = \frac{du}{dx}$$

```
    }
  b[180 + k3-2]= b[60 + k3-1];
  b[180 + k3-1]= b[     k3-2];

  b[240 + k3-1]= b[120+ k3  ];
  b[240 + k3  ]= b[60 + k3-1];

  b[300 + k3-2]= b[120 +k3  ];
  b[300 + k3  ]= b[     k3-2];
  }

return(0);
}
```

If you want to get to the bottom of the program-technical conversions with the help of the program Z88, please note the C-routines according to Table 4.7-1.

Table 4.7-1: C-routines for continuum elements

Element type	Element stiffness	Element load vector	Stress routine
20 nodes hexahedron	HEXA88.C	BHEXA88.C	SHEX88.C
8 nodes hexahedron	LQUA88.C	BLQUA88:C	SLQU88.C
6 and 8 nodes plane stress element/torus	QSHE88.C	BQSHE88.C	SQSH88.C
12 nodes plane stress element/torus	CSHE88.C	BCSHE88.C	SCSH88.C
6 nodes plate	SPLA88.C	BSPLA88.C	SSPL88.C
8 nodes plate	APLA88.C	BAPLA88.C	SAPL88.C
16 nodes plate	HPLA88.C	BHPLA88.C	SHPL88.C
10 nodes tetrahedron	TETR88.C	BTETR88.C	STET88.C
4 nodes tetrahedron	SPUR88.C	BSPUR88.C	SSPU88.C

4.8 Some Remarks on Modelling

4.8.1 Choice of Element Types

How to transform a real structure into a finite element model? One possible answer to this simple sounding, but extremely complicated question, you will find in chapter 13 with different examples. However, let us start reflecting some basic thoughts:

Since Kopernikus and Galilei we know that the world is a sphere, a 3D item and no plane stress element. The plane stress element, however, is a typical 2D item. All real components are always 3D items, so only with 3D CAD programs parts can be described really close to reality. Please keep in mind that a 2D drawing, no matter whether generated on a drawing board or in a 2D paint program, in reality only is an aggregation of drawing conventions. As a former designer's colleague of us used to say: "It is crazy! First of all, we have to flatten a real component in our head to transfer it into a drawing. Then the viewer of the drawing must rebuild the component in his head!"

Thus a part of the answer is already given: A real part can always and principally be illustrated by volume elements. This action has only one flaw: It just makes the highest demands on calculation power, main memory and disk storage. But the trend is towards this direction, and a new generation of the FE programs, which are especially intended for the designers, so to speak "for the small FE calculation in between", only operate with volume elements or shell elements. Other element types are not practically implemented any more. A typical representative of this program type is PRO/MECHANICA.

On the other hand, the classical engineering mechanics provide the typical 1D or 2D concepts, such as rope, truss, beams, torsion beams, plane stress element, plate, and membrane. However, please remind yourself that these models of the mechanics have been born from necessity, because of the equations, which describe the general spatial displacement state, the so-called *Navier's equations* (cf. */39/*) with the so-called *Lamé* constants λ and μ:

$$(\lambda + \mu)\, u_{j,ji} + \mu\, u_{i,jj} + f_i = 0$$

are only solvable analytically for very few special cases. That's why one has created the concepts for the plane, which are solvable, for example, in the case of the plane stress element with the so-called *Airy's stress function*, in the case of the plate with the *Kirchhoff's plate equation*.

It has always been the art of the structural engineer to idealize the real calculation problem. This is elegantly called "modelling" today. The reader may consult, for example, *Hirschfeld /64/*, *Mann /85/*, *Wagner/Erlhof /86–88/* and *Schnell/Gross/Hauger /89–91/*, for static problems in the civil engineering or for general interest, in case of plates also *Werkle /63/*.

How does the typical machine designer proceed? If he must enter the numerical values manually, he will try to minimize this input expenditure in any case. Hence, he will illustrate the problem as a truss work, beam framework or as a problem of the plane state of stress or the axial-symmetrical state of stress. However, if he was given a quite complicated part in a 3D CAD system, he will try to save this data in the FE program and use it further, what, in most cases, leads to volume elements and here preferably to tetrahedrons (because they can be easier generated automatically than a hexahedron).

Of course, there are structures, which one will only illustrate like this (Figure 4.8-1, example 13.2).

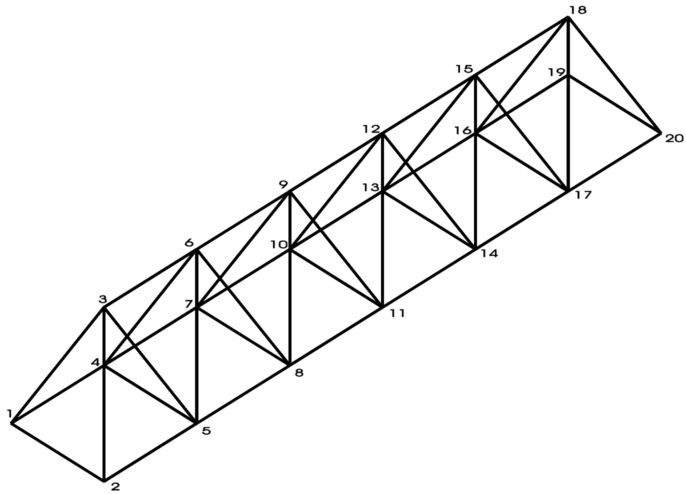

Figure 4.8-1: Crane girder: Truss- or beam framework

No reasonable person would model such a crane girder other than by a truss or beam framework. A FE structure of volume elements would not only provide very great requirements upon the computer, but would also not provide accurate results. For the gear shaft (see Figure 4.8-2, example 13.3) it depends on what you want to know. If you only want to determine the bending lines and bearing forces, you would treat this shaft like a continuous beam. However, if you are interested in the notch effect in the shaft shoulder, you could either work with axial-symmetrical elements or with volume elements, cf. example 22. Even then, only the stress concentration factors α_k can actually be calculated by FEA; to gather the micro supporting effect for the actually important *notch effect factors* β_k is very difficult.

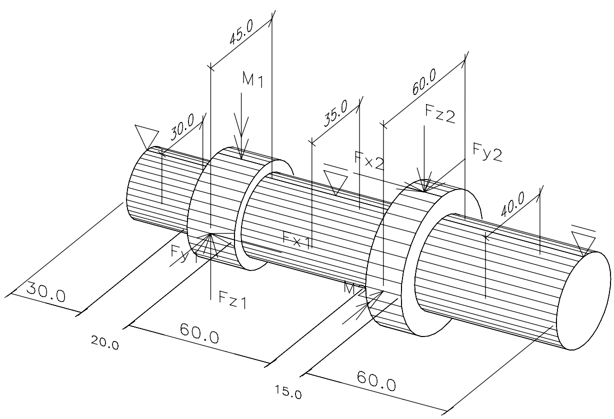

Figure 4.8-2: Gear shaft: Continuous beam

The force measurement element (Figure 4.8-3, example 13.16) is perfect for working with the plane state of stress. Volume elements would not bring better results, but more calculation expenditure. For this structure, the practitioner would decide rather how comfortable he could generate the mesh. If the meshing works well and will simply be done in the 2D case, this would be okay. On the other hand, if you can generate the mesh without additional expenditure with your 3D CAD system, then take the volume mesh, because the calculation expenditure will stay within bounds for this simple component, even with parabolic tetrahedrons.

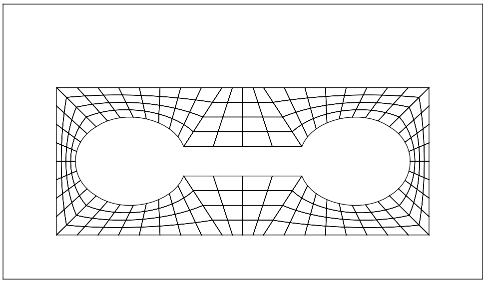

Figure 4.8-3: Force measurement element: Plane stress state

The liquid gas tank, according to Figure 4.8-4, example 13.10, also demands for a figure with axial-symmetrical torus elements. A spatial structure would be considerable, which would deliver no additional information, but require a lot more computer power.

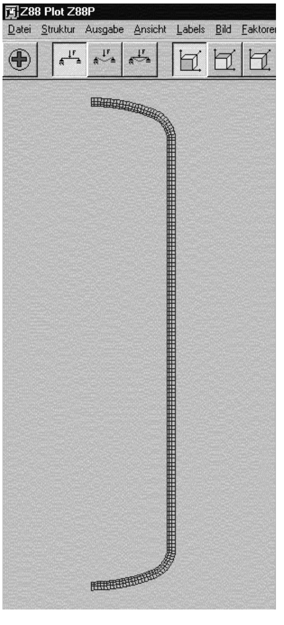

Figure 4.8 4: Liquid gas tank: Axial-symmetrical torus elements

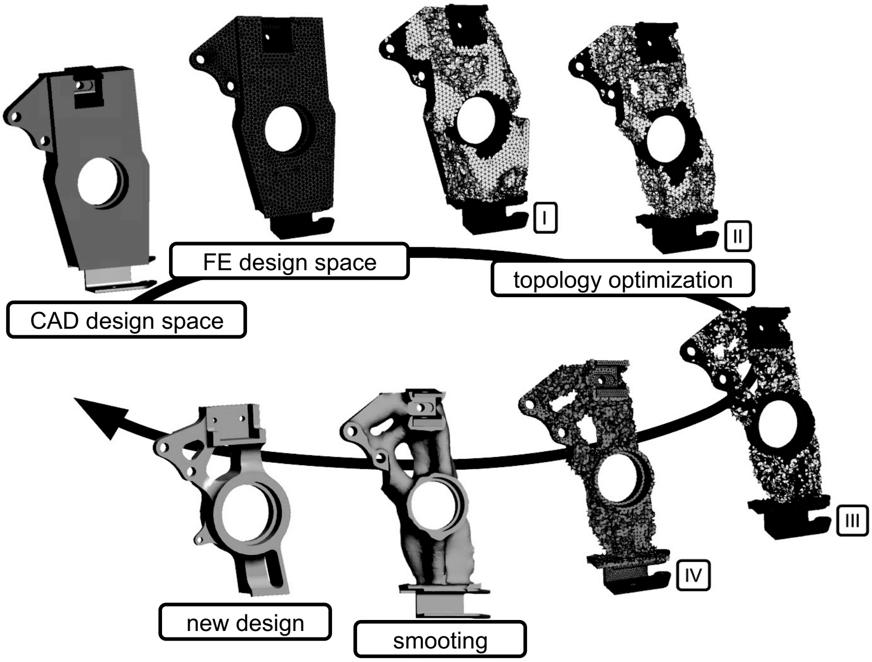

Figure 4.8-10: Development of the component geometry along the process chain

4.9 Some Remarks on Shells

This chapter shall conclude the explanations about finite elements and element stiffness matrices, and, actually, this complicated matter does not belong in an introductory textbook. Since our readers asked us for adressing shells over and over again, the lead author of this book has derived and built-in four shell elements in Z88, which are quite useful in practice. We have decided to treat the subject from a strongly simplistic view, so to speak "shells for average requirements", and to renounce detail and scientific severity. The colleagues of the engineering mechanics and the shell specialists may forgive us.

Shells are surface structures whose center surface is bent once or twice, cf. *Girkmann /113/*. The thickness is usually very small compared to the other dimensions. Analytically calculating shells is exceptionally difficult, and direct solutions have only become known for quite easy cases like rotation-symmetrical shell structures for which absolutely different basic assumptions were made, depending on the author. Furthermore, the classical shell theory makes a distinction between so-called membrane shells – the bending stresses are neglected and only the normal stresses are considered in the shell edges – and shells with bending influence.

All in all, the treatment of shells is extremely complex from an analytic viewpoint; Figure 4.9-1 shall give a first impression of the complexity. The interested reader may consult the “shell classics” like *Timoshenko/Woinowsky-Krieger /37/* and *Girkmann /113/*. From our point of view, there is a very nice work for the engineer from *Hake/Meskouris /116/*; *Pilkey /38/* offers accumulated formulae for shell problems.

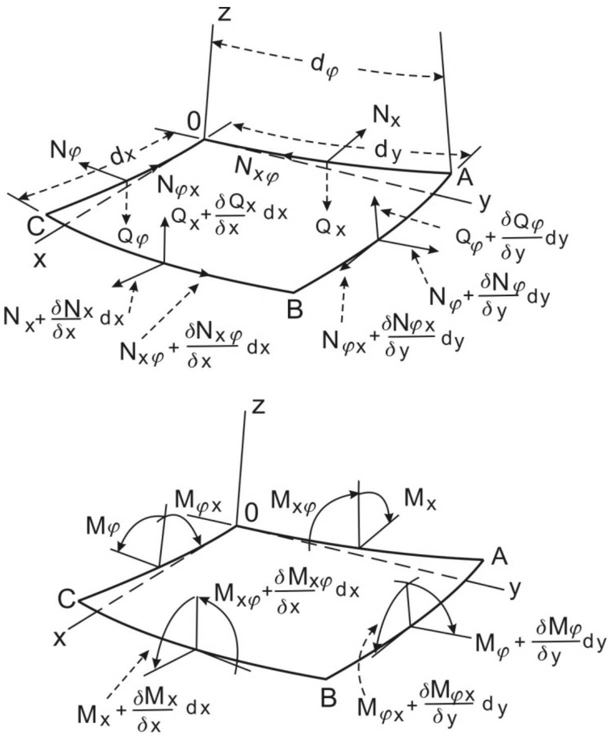

Figure 4.9-1: Deflections of a cylindrical shell, in accordance with *Timoshenko/Woinowsky-Krieger /37/*, p. 508.

The classic shell theory only helps to a limited extent for setting up element stiffness matrices, it can even mislead because it points to element forms, e.g. double curved shells which cannot be used at all by FE computer processes that are working with a CAD system. Here, we will take three other paths, and in our view, easier ways without any claim to completeness.

1. Volume Shell Elements

First it has to be made clear that nature knows nothing about shell states of stress; this is a fiction of the engineering mechanics. Hence, shells can in general be described with volume elements like hexahedrons and tetrahedrons. In many cases this also works very nicely; unfortunately, the number of elements becomes unreasonably big. In contrast to tetrahedrons, the situation is aggravated for hexahedrons because the third dimension, i.e. the thickness, is much smaller than both of the other dimensions (Figure 4.9-2).

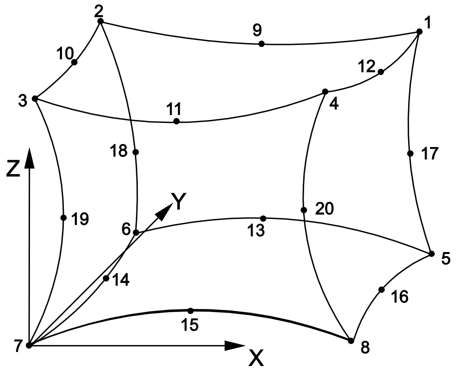

Figure 4.9-2: A hexahedron as a volume element

The shape functions for a hexahedron with 8 ~ 20 nodes generally are (cf. *Bathe /4, 5/*):

$$h_1 = g_1 - \frac{g_9 + g_{12} + g_{17}}{2} \quad , \quad h_2 = g_2 - \frac{g_9 + g_{10} + g_{18}}{2}$$

$$h_3 = g_3 - \frac{g_{10} + g_{11} + g_{19}}{2} \quad , \quad h_4 = g_4 - \frac{g_{11} + g_{12} + g_{20}}{2}$$

$$h_5 = g_5 - \frac{g_{13} + g_{16} + g_{17}}{2} \quad , \quad h_6 = g_6 - \frac{g_{13} + g_{14} + g_{18}}{2}$$

$$h_7 = g_7 - \frac{g_{14} + g_{15} + g_{19}}{2} \quad , \quad h_8 = g_8 - \frac{g_{15} + g_{16} + g_{20}}{2}$$

with $h_j = g_j$ for $j = 9 \sim 20$

$g_j = 0$ is valid if the node does not exist, otherwise: $g_j = G(r,r_j) \cdot G(s,s_j) \cdot G(t,t_j)$

Let $\beta = r,s,t$, so that

$$G(\beta,\beta_j) = \frac{1}{2}(1 + \beta \cdot \beta_j) \text{ für } \beta_j = \pm 1$$

$$G(\beta,\beta_j) = (1 - \beta^2) \text{ für } \beta_j = 0$$

To take the above considerations into account, it would make sense to only have two instead of three nodes in the axis of the thickness direction. This means a quadratic approach in two axes and in the third axis a linear displacement approach. The respective code paragraph of the volume shell routine `SHAQ88.C` demonstrates this:

```
/*----------------------------------------------------------
* Setting the brackets of the shape functions
*----------------------------------------------------------*/
rp=  1. + (*r);
sp=  1. + (*s);
rm=  1. - (*r);
sm=  1. - (*s);
```

9 Z88: The Basics

9.1 General Information

Two free programs form the basis of this book: Z88V14 Open Source and Z88Aurora V2. While the Open Source version works especially basis-oriented and “originally”, at which you, with the help of the provided source code, can understand all theoretical basics of the previous chapters, our new development Z88Aurora provides a very comfortable user interface, which compared to Z88V14 Open Source perceptibly facilitates the FE calculation in everyday studies but also in industrial practice. While you work directly with input files and output files in the Open Source version, these files (which are identical for Open Source version and Z88Aurora) are automatically created by Z88Aurora, and you can, e.g., very comfortably apply boundary conditions.

All static linear examples of this book of course can be calculated with both Z88 versions, and it is up to you to decide which version you use to work through the examples. If you want to work very comfortably from the beginning on and if you are less interested in the program backgrounds, you should choose Z88Aurora V2, with which the use is very similar to commercial programs. If you want to know everything in detail, you are not annoyed by a relatively simplistic user interface and you want to study or even change or extend the program code, then give Open Source Z88V14 a try.

As you know, both program versions come from our department: Z88V14 Open Source and Z88Aurora V2 are at the moment respectively the newest versions, and in the internet you will find revised and actual releases under *www.z88.de*.

9.1.1 Summary of the Z88 Element Library

Two-dimensional Problems: Plane Stress Elements, Plates, Beams, Trusses

Plane stress element no. 3

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

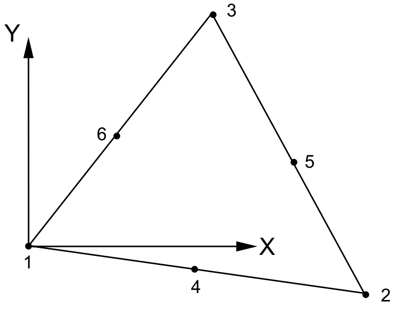

Figure 9.1-1: Plane stress element no. 3

Plane stress element no. 7

- Quadratic isoparametric Serendipity element
- Quality of displacements: very good
- Quality of stresses in the Gauss points: very good
- Quality of the stresses in the corner nodes: good
- Computing effort: high
- Size of element stiffness matrix: 16×16

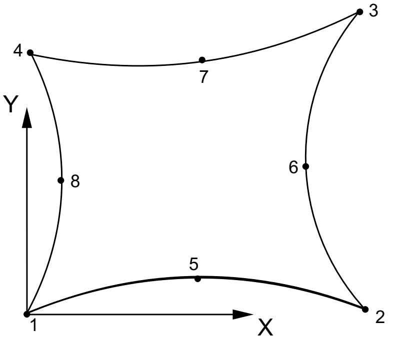

Figure 9.1-2: Plane stress element no. 7

13 Examples

In this chapter, 32 examples (with other sub examples) are covered, of which the examples 1 to 24 can be carried out with the Open Source version of Z88 as well as with the Z88Aurora Freeware. The examples 25 to 32 are especially designed for the use of Z88Aurora. For all examples, the respective input files are in the correspondent directories “Z88 V14” and “Z88Aurora” on the DVD. Only the calculation must be carried out. You can import all files of Z88V14 in Aurora; as it is explained later. The examples 4, 6, 7, 13, 17 and 18 can easily be analytically recalculated.

The first examples are described very detailed and step by step, so that you can get familiar with the procedure very quickly. Then the later examples require the knowledge of how to use Z88, and they concentrate more on the background of the respective job. The first examples are easy, then they gradually become more complicated, hence, you should work through the examples in sequence.

If examples do not start, a memory problem can be on hand. Do other programs require memory, particularly these fat and greedy memory eaters like office packages? All examples were tested on the different computer systems and operating systems, and the smaller examples run even on older computers. Current PCs compute very big Z88 structures without any problems as shown in example 21. The biggest computed structure up to now had 12 million degrees of freedom and ran on a 64 bit PC with 64 bit Windows or with 64 bit LINUX. Adjust Z88.DYN if necessary. Mind the .LOG files: If the memory does not suffice, it is noted in this file.

After you have tried the prepared examples, you should design your own examples in your CAD system. Export your models/drawings with a CAD system compatible to AutoCAD as DXF files and convert them with Z88X to Z88 files for Z88 V14 OS or to a STL or STEP file for Z88Aurora. If the Z88-DXF converter Z88X does not convert your DXF files properly, then particularly repeat the steps 3 and 5 of the chapter 10.7.2. If nothing works, try another CAD program.

Do you have a 3D CAD system with an integrated automesher? Then you can export FE meshes in ANSYS-PREP7, COSMOS or NASTRAN format and convert them to Z88 input files with Z88G or Z88ASY in Z88 V14 OS or with the import function in Z88Aurora.

Tips for Z88 V14:

The import and export files are shown partially shortened so to not fill sides needlessly. Only the essentials should be shown. You can start all examples any time yourself.

Furthermore, consider the protocol files .LOG generated by the Z88 modules. Vary the input files, especially the mesher input files of examples1, 5 and 7. Thus, one gets a feeling for the use of Z88.

Note that the floating point Figure 0 is never really a zero in a computer, but is shown as an approximation. Hence, even the input, which is given in Z88I1.TXT as 0 can reappear in output file Z88O0.TXT in very small figures, caused by formatting of the run time system. This is normal. This applies even more for calculated values, as for example displacements in Z88O2.TXT, stresses in Z88O3.TXT and nodal forces in Z88O4.TXT. Such values are always to be seen relatively to other values: Is the biggest calculated displacement in Z88O2.TXT for example 0.1 mm, then another displacement with e. g. 1.234E-006 mm should be regarded as de facto zero.

In ANSYS-PREP7, COSMOS or NASTRAN files first check with Z88R in the test mode how much memory is required and how good the node numbering is. If necessary, run the Cuthill-McKee algorithm Z88H in Z88 V14 OS. But it is better to use one of the sparse matrix solvers from the beginning.

Tips for Z88Aurora V2:

The examples exist as a complete project directory as an overview or the import files can be used to generate your own examples. The process of the FE analysis in Z88Aurora always proceeds according to the workflow that can be seen in Figure 13.0-1.

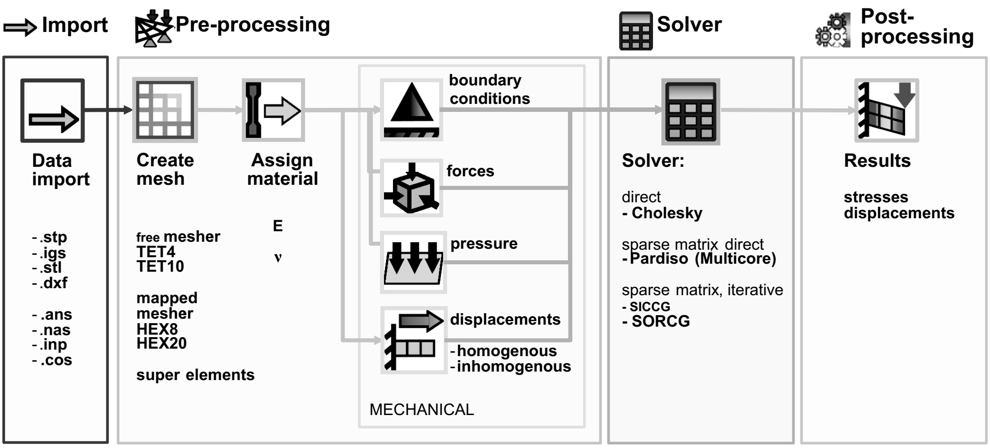

Figure 13.0-1: Proceeding of finite element analysis with the example of a static mechanical analysis

Z88Aurora distinguishes itself by the intuitive operation of the pre-processor and post-processor. The project data management takes place with a project folder management. A status display provides better ease of use. Several menu bars are important for the operation. Four icon menu bars offer the fast access of all functions of Z88Aurora. The main functions of the first icon menu bar, as for example pre-processor, open additional side menus. The other three icon menu bars contain various view manipulations, colour settings, import options and the pre-processor functionalities.

13.25 Submarine made of Shells No. 22

A submarine („U-boat“) of class 212A of the German Navy, which was constructed as a shell structure in Pro/ENGINEER is imported into Z88Aurora with the help of NASTRAN and thickened to a volume shell. We calculate the deformation and the stresses of the submarine body at a diving depth of 50 meters. The submarine is in a state of poise in the water. This is why we fix it in Z88Aurora with a virtual fixed point, “floating” in the space.

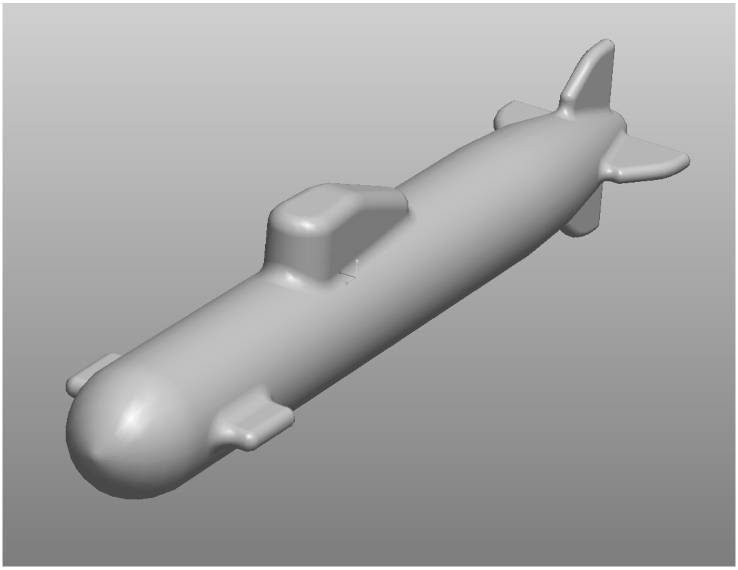

Figure 13.25-1: Geometry of the submarine designed in Pro/ENGINEER

Create a new Project Directory

Create a new project directory .

Import NASTRAN

The example file *u-boat.nas* from *z88_examples_z88aurora/b25/Nastran-File* is imported as a NASTRAN file with the import object menu. Select the import option “shell”.

Modelling the FE Structure out of Superelements

In the next step, we want to mesh the conventional shell structure of the submarine to volume shells. Switch to the *Pre-processor menu* → *Super elements*. The volume shell structure should be 20 mm thick:

1. Set thickness: Value “20”.
2. Administration: “Add” the new meshing rule.
3. Generate FE structure: “Create mesh”.

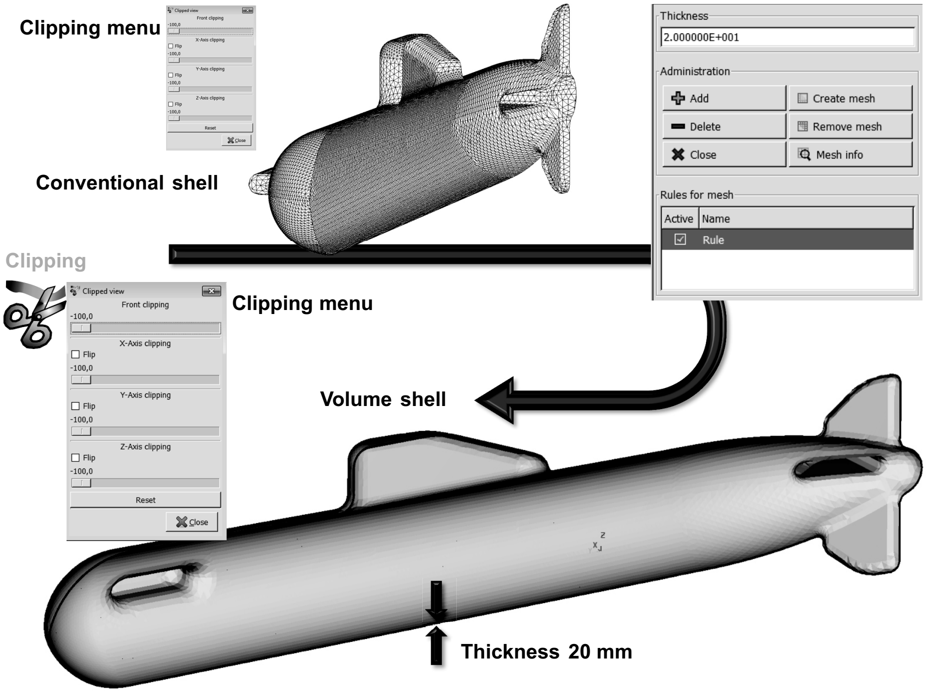

Figure 13.25-2: Modelling the volume shells

Clipping

With the help of the clipping function we can control that the conventional shell structure has been thickened to volume shells.

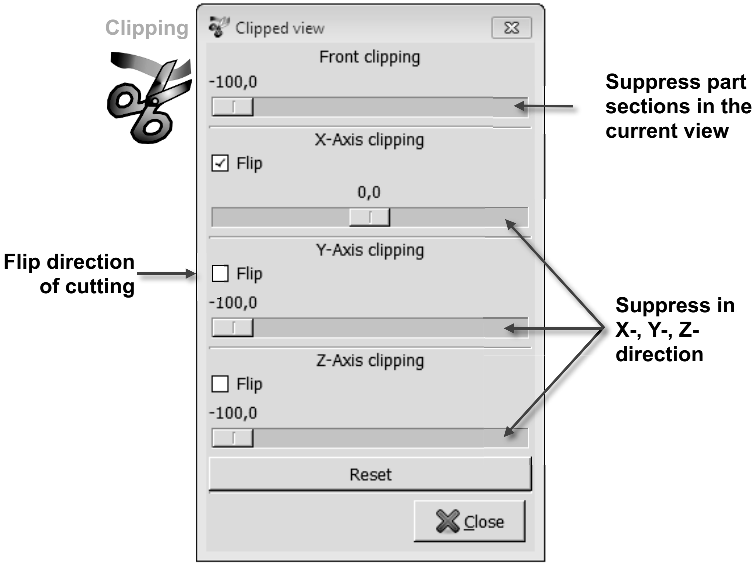

Figure 13.25-3: Clipping menu

Assigning the Material

Use structural steel S235JR from the Z88Aurora material database.

Surface picking – Node picking

Switch to the "picking context menu" and "node picking" and set to node sets, called "X_direction" and "Z_direction", for the virtual fixed point. Besides, the surface set "shell surface" has to be created. This surface set represents the whole exterior surface of the submarine and contains the boundary condition of pressure. Select a surface facet in the context menu "surface picking", put the slider for selecting the "angle" to value "50" and pick the whole exterior surface by using the button "surface".

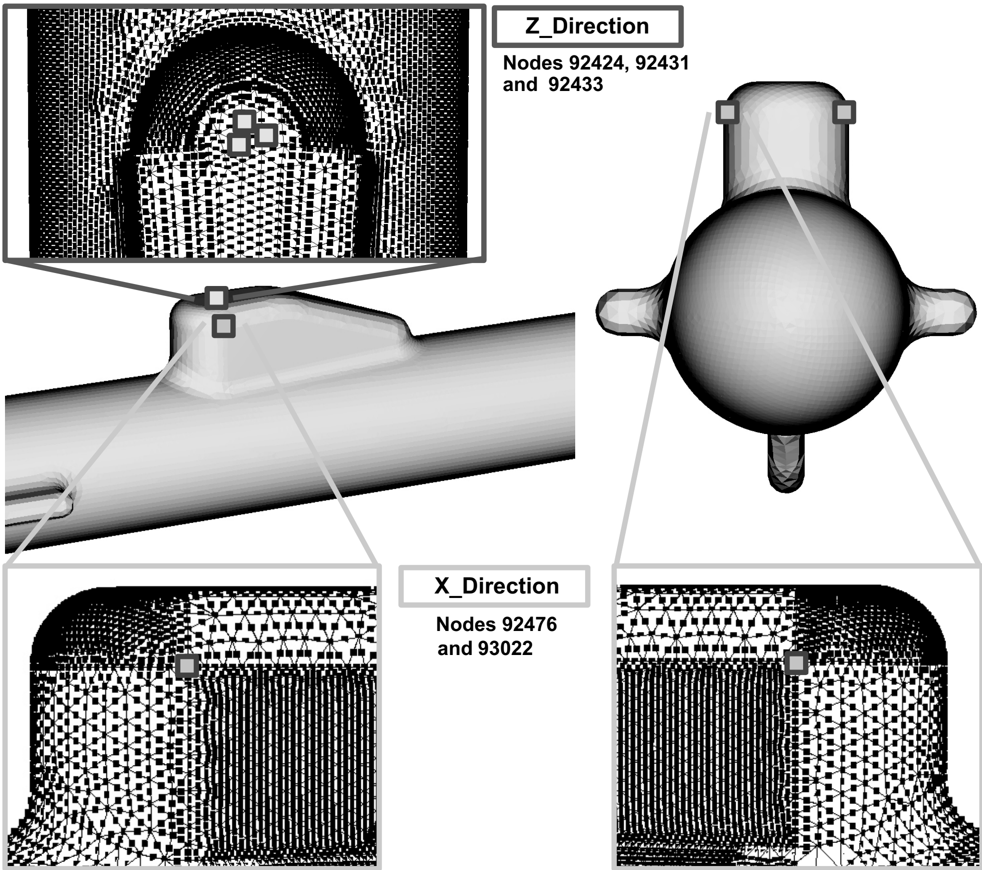

Figure 13.25 4: Node sets for virtual fixed point

Boundary Conditions

Button *Pre-processor* → *assigning boundary conditions* → In the context menu, we assign boundary conditions to the node sets and the surface set. A hydraulic pressure of 0.5 N/mm^2 is set to the whole shell surface. The node sets are fixed in a way that guarantees statically defined support but lets the submarine "freely" flow in the water.

1. Support: Set “Z_direction”, direction X, Y, “displacement”, value “0”, name “XY_locating_support”.
2. Support: Set “X_direction”, direction Y, Z, “displacement”, value “0”, name “YZ_locating_support”.
3. Pressure: Set “shell surface”, pressure, value “0.5”, name “hydraulic_pressure”.

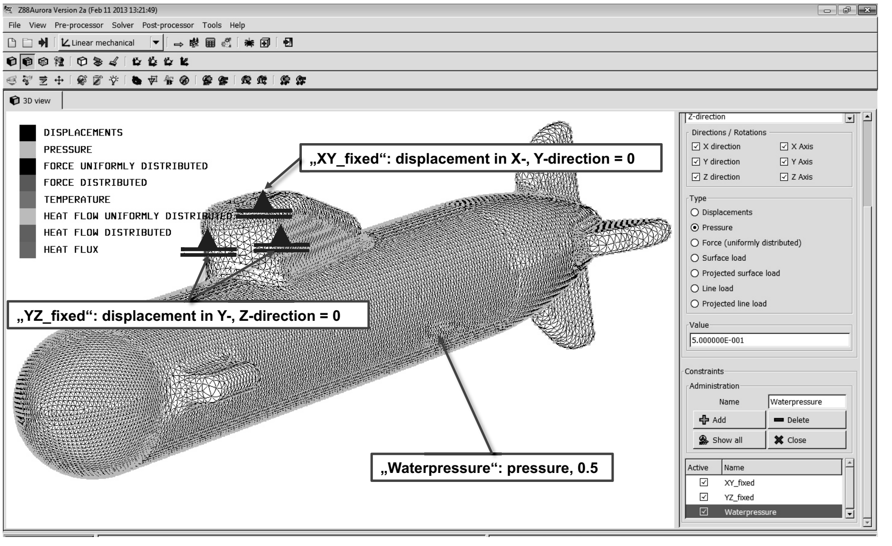

Figure 13.25-5: Boundary conditions

Launching the Calculation

Start the calculation with the “PARDISO solver”.

Outputs

The PARDISO solver delivers following displacements and stresses in the corner nodes:

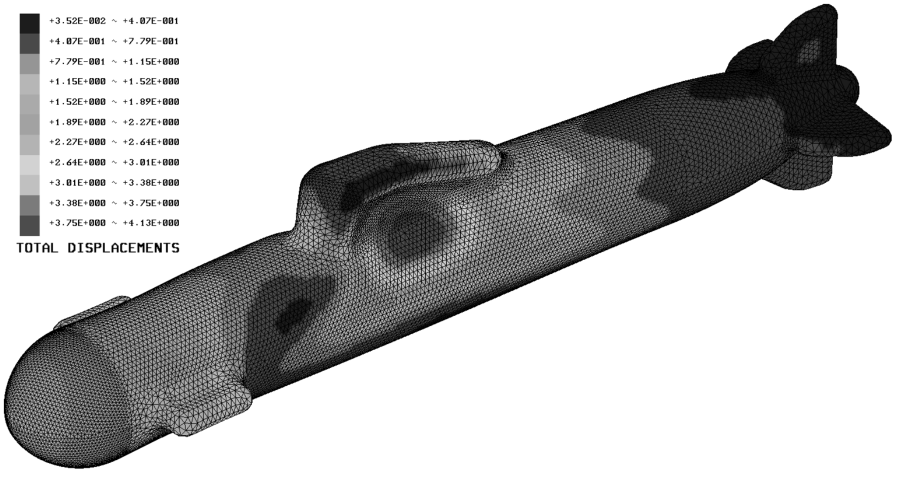

Figure 13.25-6: Display of the results: displacements

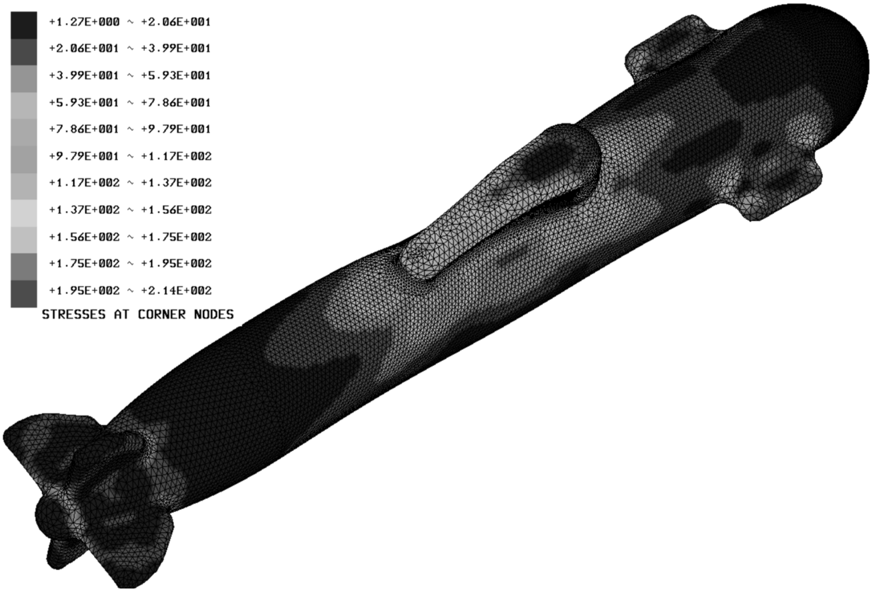

Figure 13.25-7: Display of the results: reduced stresses in the corner nodes according to von Mises

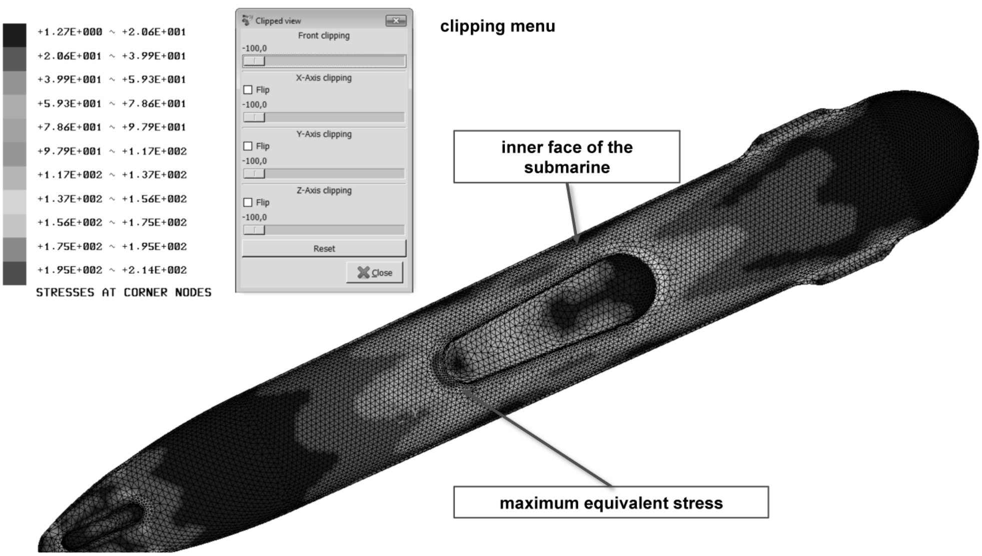

Figure 13.25-8: Display of the results inside the submarine: reduced stresses in the corner nodes according to von Mises