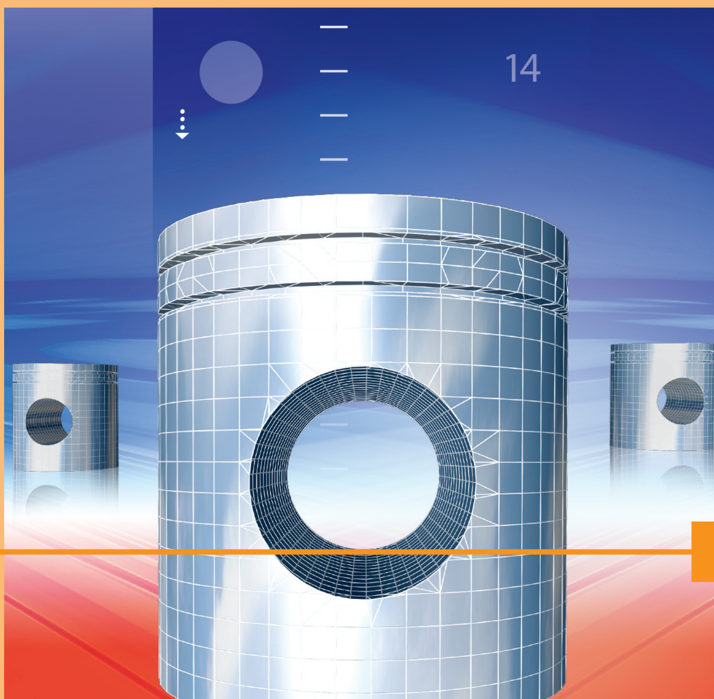
Frank Rieg
Reinhard Hackenschmidt
Bettina Alber-Laukant

Finite Element Analysis for Engineers

Basics and Practical Applications with Z88Aurora



HANSER

Frank Rieg
Reinhard Hackenschmidt
Bettina Alber-Laukant

Finite Element Analysis for Engineers

Frank Rieg
Reinhard Hackenschmidt
Bettina Alber-Laukant

Finite Element Analysis for Engineers

Basics and Practical Applications with Z88Aurora

HANSER

The Authors:
Prof. Dr.-Ing. Frank Rieg, Full Professor, Chair for Engineering Design and CAD, University of Bayreuth
Dipl.-Wirtsch.-Ing. Reinhard Hackenschmidt, Senior Academic Councillor, Chair for Engineering Design and CAD, University of Bayreuth
Dr.-Ing. Bettina Alber-Laukant, Patent Scientist, Academic Councillor, Chair for Engineering Design and CAD, University of Bayreuth

Translated by the authors with the help of Franziska Auer, Teresa Bertelshofer, Kevin Deese, Christoph Gürtner and Marlene Süß

Distributed in North and South America by Hanser Publications
6915 Valley Avenue, Cincinnati, Ohio 45244-3029, USA
Fax: (513) 527-8801
Phone: (513) 527-8977
www.hanserpublications.com

Distributed in all other countries by Carl Hanser Verlag
Postfach 86 04 20, 81631 Munich, Germany
Fax: +49 (89) 98 48 09
www.hanser-fachbuch.de

Cataloging-in-Publication Data is on file with the Library of Congress.

Bibliografische Information der deutschen Bibliothek:
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

This book has an accompanying DVD. If the DVD is not enclosed, it can be ordered free of charge by sending an e-mail request to fachbuch@hanser.de.

Copyediting: Olivia Brand
Production Management: Der Buchmacher, Arthur Lenner, Munich
Coverconcept: Marc Müller-Bremer, www.rebranding.de, Munich
Coverdesign: Stephan Rönigk
Cover illustration: Atelier Frank Wohlgemuth, Bremen
Typeset: le-tex publishing services GmbH, Leipzig
Printed and bound by Kösel, Krugzell
Printed in Germany
ISBN 978-1-56990-487-9
E-Book ISBN 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.

1 Introduction

Many approaches in technology are centuries-old. The elasticity theory, for example, was practically developed as a whole in the 19th century. The so-called method of the finite elements though originated not until the upcoming of the first digital computers in Germany, in the United States and in England during World War II. These first computers, the German Zuse Z3 of 1941, but particularly the American Harvard Mark I, served for the calculation of trajectories for the artillery (cf. */33/*). At the same time, a new model of airplane came up, the jet-propelled airplane. Its yet unknown speed led to entirely new problems – new air foil designs like the swept-back wing, extremely light and still very stable cells, which do not fail at big heights, and the jet engines themselves.

Hence, it did not come by chance that in the 50s, *J. Turner* and *R. Clough*, engineers at Boeing in Seattle developed the matrix force method and the matrix displacement method for the static calculation of cells and wings. Already at the end of the 40s, *J. Argyris* from England showed that one can describe continuums in simplified form by disassembling them in smaller subareas. Before these considerations, *Hrenikoff* suggested disassembling continuums in an array of trusses or beams to solve stress and plate problems; this was called "Framework method" in the literature. The first one, who used the concept of the Finite element method publicly at a conference, seems to have been *R. Clough* in 1960.

We already indicated that originally there was a *matrix force process* and a *matrix displacement process*. While in the matrix force process the wanted unknowns are the forces – actually, this is the approach, which is also common in the classical engineering mechanics – the unknowns in the matrix displacement process are the displacements of the system, which is at first sight rather unusual. It was argued for a long time, in practice and science, which of both approaches is the better one. Today this question has been answered: All large program systems exclusively work according to the displacement process because it can be schematized and programmed much easier and straight forward.

In the early days, only a few "privileged" people could generally execute finite elements calculations, because only they had access to big computers which was completely unaffordable for most universities and companies at that time. When the lead author of this book wrote his diploma thesis (i.e. master thesis) in 1978, the calculation of a race car body for the car manufacturer Porsche, done with the finite element method, at least one useful FE program, *SAP IV* (Structural Analysis Program) of *Wilson* and *Bathe*, was available. It ran in batch-mode on the very large IBM 370/168 computer equipment of the University of Technology Darmstadt, Germany. The input data was not entered by a terminal, but on punch cards which one had to punch with an IBM cardpunch. After much ado, the input record was finally punched, and one could carry the card deck to the computer center. At night, and only at night because of the "huge" core memory requirement of ~700 Kbyte, *SAP IV* ("Structural Analysis Program") was started and one could eventually fetch the results the next day on centimeter thick paper piles. Interactive graphics? Absolutely unknown. At least plotting on paper was possible, but in addition, another program *SAPOST* was necessary, which got its plot instructions through punch cards, too.

This has changed long ago, and if one looks at pictures of mainframes even of the early 80s today, one could think, these photos are shot on another planet. Especially the *personal computer* has performed great pioneering work in the 80s. Already in the middle of the 80s, one could calculate quite substantial FE structures with PCs (cf. */27/*), but the limit at that time was a usable main storage of about 500 Kbyte, dictated by DOS. In 1985, the lead author of this book started with the development of his FE program Z88 on an IBM AT, at that time still as a FORTRAN version (cf. */28/*). In the beginning of the 90s, the triumphal success of Windows with the version 3.0 stood out, and the lead author now re-coded the Z88 program completely in C (cf. */15/*) because at that time one could provide real Windows programs only with the computer language C.

Today, everybody knows the situation: Every cheap PC from the discounter has much more power – and that means several powers of ten – than the IBM 370 mainframe from more than 30 years ago, and nowadays everybody can carry out extensive finite elements calculations on his/her PC at home.

As we will see in the second chapter, the method of the finite elements – or better *finite element analysis* according to our opinion, because one checks and analyses something in contrast to the methodical design which is a synthesis – is in principle exceptionally simple! The special thing about it actually only is the strictly formalized proceeding what is indeed very suitable for computer usage. When introducing the basic action in chapter 2 we deliberately do this from an elastostatic point of view and first only work with trusses and beams. However, trusses and beams are of course no 2D or 3D continuums, and some readers will find this approach dilettantish.

But stop – to show the basic action of the *matrix displacement process*, these simple elements are really very suitable. Since finite elements for plane stress, for plane strain, for axisymmetric stress, for plate bending and for spatial stresses – to mention the most important ones – are completely integrated into the process! In fact, all computer routines for putting up the element stiffness matrices are constructed quite alike, as you can check any time with our book with the help of the enclosed program procedures for Z88 in C. Compare, for example, the subroutine *SHEI88.C* for curvilinear 8 nodes Serendipity plane stress elements and axisymmetric elements with the routine *HEXA88.C* for curvilinear 20 nodes Serendipity hexahedrons for the general spatial state of stress.

By using the term matrix displacements processes one recognizes, by the way, all relevant aspects: We deal with partly gigantic *matrices*, *displacements* are calculated, namely with a schematized *process*.

You may already have guessed it: One can approach this process either from the engineer's side, as we will do in chapter 2, or from the strictly mathematical side. Which way one selects, certainly depends on his/her education and previous experience, but also depends on which aim one wants to pursue. Because the process of the finite elements was developed by engineers for solving engineering problems, we find it appropriate to derive the basics also from the engineer's point of view. Moreover, this has the advantage that the reader only needs to have basic knowledge of the matrix calculation in addition to the basic math skills of a high school graduate. However, what is absolutely necessary, is solid knowledge in the area of "rigid" statics and elastostatics. Who is not well-versed in these, will fall flat on his face very soon when working with any FE program, not only with Z88 or Z88Aurora, and eventually even break his neck. Why? Because there are two obstacles when working with the finite element analysis, and they are system-immanent. The first trap: the actual generating of the finite element mesh (how rough or how accurate, which element types) – this takes a lot of experience and training. The second trap: the choice of the boundary conditions, that means when and where to attach fixtures, apply forces and the like. Here, experience is also needed, but first of all solid mechanics knowledge is required: Even with the most expensive computer program a statically under-determined system breaks down in itself.

But also for engineers approaching the finite element analysis from a strictly mathematical view can be absolutely exciting and does make sense. Indeed, elastostatic problems can be described by extremum principles, e.g., by the principle of the minimum of the whole potential energy: Amongst all displacement states, which fulfill the kinematic boundary conditions, the actual equilibrium minimizes the potential energy. These functionals of the potential energy, which can be put up for trusses, beams, torsion trusses, plane stress elements, plates etc. must become stationary. This can be done with the method of *Ritz*. The functions of Ritz's procedure are definitely related to the approximation or form functions of the finite element analysis. One of the prominent sources for the derivation of the various element stiffness matrices with functionals is the book of *Schwarz /6/* which can be very much recommended to the mathematically interested reader.

However, we would like to remind you again of the fact that one cannot study the finite element analysis only by theoretical considerations. Only by extensive training and work on the computer, one will bring it to certain mastery in this area. It seems important to us that you understand, sensibly change and complement the examples of the book, and that you do this on the computer. Hence, we have kept the theoretical chapters relatively short, so that you get to see the practical aspects as soon as possible. However, whenever we believed that certain questions, e.g., the elasticity theory, are not given their fair share in other literature, we intentionally haven't accepted abridgements on this topic in our book.

Practicing with the computer, which is very much recommended by us, can be carried out with Z88 Version 14 and Z88Aurora version 2, both programs being provided by us. Z88 is used in this book since the first edition and comes with all C program sources as well as *nmake* files for Windows Visual Studio 2008 and *make* file for LINUX and Mac OS X. This way, you can study the source code and extend it if necessary or change it, which really makes sense if one wants to understand the theoretical foundation of the FEA. In order to purely work through our many examples, you, of course, do not need to deal with the internal matters of programming. Z88 is ready-to-run for all different operating systems on the DVD. A manual of approximately 200 pages in PDF format comes with it, from which you can obtain tips for the installation and operation. For a quick start you find an edited version of the operating instructions in chapters 9 and 10. Z88 is a quite classical and original FE program, which is controlled through input and output files. As with every classical program of this type it is very suitable to study basic actions, but for everyday work it is a little bit clumsy and not really comfortable to use.

Hence, in 2009 our former employees Bernd Roith, PhD, Alexander Troll, PhD, and Prof. Martin Zimmermann, PhD, had the idea to create a very contemporary operating surface with the name *Z88Aurora* on top of the structures of Z88, whose special focus is to directly read in CAD files in the STEP or STL format, to generate the mesh and to be able to provide it interactively with boundary conditions and materials. Very quickly, other employees of the department joined the work. In June 2010, the first version Z88Aurora was released and immediately was a complete success. In June 2012 Z88Aurora V1 was downloaded about 32,000 times worldwide on our Internet site! Soon after, Bernd Roith, PhD, began with the work on the next version. His successor was Markus Zimmermann, PhD, who extensively formed the system in its current version 2, together with B. Alber-Laukant, PhD, M. Frisch, M.Sc, M. Neidnicht, PhD, F. Nützel, M.Sc, C. Wehmann, PhD, J. Zapf, PhD, and Prof. F. Rieg, PhD. You will experience this new version 2 of Z88Aurora as very pleasant and intuitive to use, to which also a unique feature contributes, the so-called *Spider Help* of B. Alber-Laukant, PhD, and Bernd Roith, PhD. *Spider Help* leads you through the workflow of a finite element analysis.

It was immediately clear that we must integrate this new development of Z88Aurora into the fourth German edition (and into this first English edition) of our FEA book. Although we have submitted Z88Aurora V2, which you find on the DVD, to very extensive tests, you all know that software can never be perfect. If you discover errors or irregularities, do not hesitate to inform us or consult our website *www.z88.de*, in order to check whether there are any program updates available.

And now: Let’s begin with the *basic procedure*!

2 The Basic Procedure

We will now briefly present the basics of the *finite element analysis* and for reasons of simplification consciously assume several issues instead of establishing them. Here you will obtain an overview and see the easiness of the procedure after reading a few pages. We do not want to conceal the fact that these issues are related to complex theories and mathematical procedures, but we will consider this in chapters 3 to 8, after we gained a total overview – actually you can skip these chapters while reading it for the first time. If you follow our explanations of the next pages, you will understand the principles of *finite element analysis*. The following are only refinements and special aspects. Wouldn't you say this is highly motivating?

At the beginning we observe an easy tension spring of steel that is clamped at one end and loaded at the other. We can carry out the loading process in two different ways: Either we apply a known force of, for example, 100 N or we drag the spring a defined distance from, for example, 5 mm. The spring is subjected to Hooke's law $F = K \cdot U$, that means, the spring force F is the product of the spring stiffness K and the spring way U.

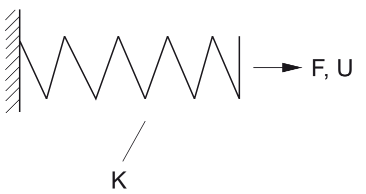


Figure 2-1: Hooke's law

The majority of everyday objects own the same properties as this screwing spring, i.e. they deform in a linear-elastic way: The force F and the distance U are proportional to each other. Every random small force causes a distance, a displacement or a deformation. This is why a rope or a tensile bar with the length ℓ, the profile A and the Young's modulus E has the following force displacement relation:

$$F = \frac{E \cdot A}{\ell} \cdot U$$

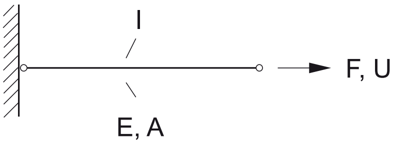


Figure 2-2: The tensile bar

If you set:

$$K = \frac{E \cdot A}{\ell}$$

then one recognizes Hooke's law again

$$F = K \cdot U$$

Now we define a truss by setting the deformation U_1 resp. the force F_1 to its left ending and the deflection U_2 resp. the force F_2 to its right one:

Figure 2-3: The in general defined truss

If we form the force equilibrium state, we receive:

$$F_1 = K \cdot U_1 - K \cdot U_2$$
$$F_2 = K \cdot U_2 - K \cdot U_1$$

Displaying this equation set in matrices notation leads to:

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

Proof: By multiplying, one receives:

$$K \cdot U_1 - K \cdot U_2 = F_1$$
$$-K \cdot U_1 + K \cdot U_2 = F_2$$

We will call the expression

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} = \begin{bmatrix} EA/\ell & -EA/\ell \\ -EA/\ell & EA/\ell \end{bmatrix}$$

element stiffness matrix. This applies to a horizontally lying truss in plane. The system of equation is written in usual matrices notation. We write the same in symbolic matrices notation (matrices and vectors in symbolic representation from now on are written in ***bold-italics***):

$\boldsymbol{K\,U = F}$

This is again Hooke's law, but this time with matrices instead of scalars.

Arithmetic example 1

We set the forces F_1 and F_2 on a truss:

Figure 2-4: Forces on a truss

with

$F_1 = -1{,}000$ N
$F_2 = +1{,}000$ N

The truss has following specific values:

length $\ell = 1{,}000$ mm

Young's modulus $E = 200{,}000$ N/mm^2

Cross sectional area $A = 100$ mm^2

K becomes:

$$K = \frac{200{,}000 \cdot 100}{1{,}000} = 20{,}000 \text{ N/mm}$$

Inserting the values reveals:

$$\begin{bmatrix} 20{,}000 & -20{,}000 \\ -20{,}000 & 20{,}000 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} -1{,}000 \\ +1{,}000 \end{bmatrix}$$

Multiplying the system of equation:

$$\begin{aligned} 20{,}000 \cdot U_1 - 20{,}000 \cdot U_2 &= -1{,}000 \ (\text{B}1) \\ -20{,}000 \cdot U_1 + 20{,}000 \cdot U_2 &= +1{,}000 \ (\text{B}2) \end{aligned}$$

To solve the 2×2 equation system, we add the equations (B1) and (B2), to eliminate one of the two unknowns:

$0 + 0 = 0$

The result of the addition is correct, however, it delivers no solution. Why? Because the equations are linearly dependent! For example: multiplying equation (B2) with −1 delivers (B1). When do such things appear? *When a system is statically underdetermined!*

1ST RULE OF FEA

Never statically under-determined, always statically defined (kinematically determined) or arbitrarily statically over-determined!

Defining a boundary condition:

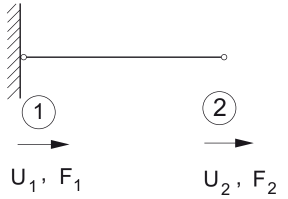


Figure 2-5: If a fixed support is in the point 1, then the displacement $U_1 = 0$

F_1 cannot be applied as an external force anymore, because the support intercepts everything! Only the following can be applied:

U_2 a displacement or

F_2 an external force

We have to fundamentally differentiate addressing this task: What kind of solutions are we looking for, forces or displacements? The already known system of equations

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \qquad \text{resp.} \qquad \boldsymbol{K}\,\boldsymbol{U} = \boldsymbol{F}$$

would be like this in mathematics: $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$.

This is the usual representation of a linear equation system: $\boldsymbol{A}$ is the coefficient matrix, $\boldsymbol{x}$ is the solution vector, i.e. the unknowns, and $\boldsymbol{b}$ is the right side. Hence:

Default of the external forces and calculating the displacements
= **Displacement-based procedure**

There is also another path to solve the equation system:

$$\boldsymbol{K}^{-1}\boldsymbol{F} = \boldsymbol{U}$$

$$\boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{x}$$

$\boldsymbol{A}^{-1}$ is the inverse of $\boldsymbol{A}$. Hence:

Default of the displacements and calculating the forces
= **Force-based procedure**

Today nearly all of the FEA systems use the displacement-based procedure:

2ND RULE OF FEA

FEA = Calculating the displacements of the structure

Our task was:

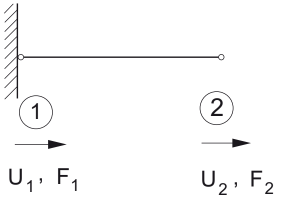


Figure 2-6: Fixed support in point 1

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

K is the structure stiffness matrix, ***U*** represents the displacements, i.e. the unknown quantity of the system, and ***F*** represents the external forces.

The structure stiffness matrix ***K*** corresponds to the element stiffness matrix $\boldsymbol{K}^{\mathbf{rod}}$, since there is only one single element, the rod.

The boundary condition is: $U_1 = 0$, a so-called *homogeneous boundary condition!* These homogeneous boundary conditions are considered in the equation system as follows:

METHOD 1: INSTALLATION OF THE HOMOGENEOUS BOUNDARY CONDITION $U_j = 0$

m1.1: In ***K***, set row *j* to 0

m1.2: In ***K***, set column *j* to 0

m1.3: In ***K***, set diagonal element *j* in to 1

m1.4: In ***F***, set force F_j to 0

Hence:

m1.1 and m1.2

$$\begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

m1.3

$$\begin{bmatrix} 1 & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

m1.4

$$\begin{bmatrix} 1 & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} 0 \\ F_2 \end{bmatrix}$$

The calculation on proves:

$$1 \cdot U_1 + 0 \cdot U_2 = 0 \rightarrow U_1 = 0$$
$$0 \cdot U_1 + K \cdot U_2 = F_2 \rightarrow U_2 = \frac{F_2}{K}$$

How do we calculate the actual rod forces, i.e. the internal forces? Until now we only considered the external forces.

METHOD 2: NODAL FORCES CALCULATION

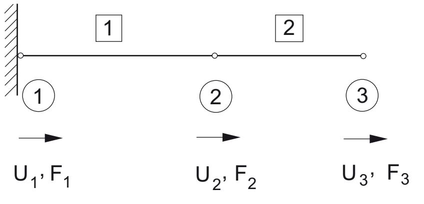
m2: Multiply the respective element stiffness matrix of the particular element with the calculated displacements.

Hence:

$$\begin{bmatrix} K & -K \\ -K & K \end{bmatrix} \begin{bmatrix} 0 \\ F_2/K \end{bmatrix} = \begin{bmatrix} -F_2 \\ F_2 \end{bmatrix}$$

These are the rod forces in the element, i.e. the internal forces. *Actio = reactio!*

Arithmetic example 2



$F_1 = 0$
$F_2 = 0$
$F_3 = 5{,}000$ N

Figure 2-7: Example with two rods

Truss 1: $\ell_1 = 500$ mm, $E_1 = 206{,}000$ N/mm^2, $A_1 = 100$ mm^2
Truss 2: $\ell_2 = 400$ mm, $E_2 = 206{,}000$ N/mm^2, $A_2 = 40$ mm^2
Hence: $K_1 = 41{,}200$ N/mm, $K_2 = 20{,}600$ N/mm
With this, the element stiffness matrices are as follows:

$$\text{first rod} = \text{FE}_1: \begin{bmatrix} K_1 & -K_1 \\ -K_1 & K_1 \end{bmatrix} = \begin{bmatrix} 41{,}200 & -41{,}200 \\ -41{,}200 & 41{,}200 \end{bmatrix} = \boldsymbol{K}_1^e$$

$$\text{second rod} = \text{FE}_2: \begin{bmatrix} K_2 & -K_2 \\ -K_2 & K_2 \end{bmatrix} = \begin{bmatrix} 20{,}600 & -20{,}600 \\ -20{,}600 & 20{,}600 \end{bmatrix} = \boldsymbol{K}_2^e$$

Both element stiffness matrices have to be combined to a structure stiffness matrix. We call this process *compilation*.

We have:

$$\boldsymbol{K} = \sum_i \boldsymbol{K}_i^e$$

3RD RULE OF FEA

Structure stiffness matrix = sum of the element stiffness matrices

Here:

Element 1 **Element 2**

$$\begin{bmatrix} K_1 & -K_1 \\ -K_1 & K_1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \qquad \begin{bmatrix} K_2 & -K_2 \\ -K_2 & K_2 \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \end{bmatrix}$$

The structure stiffness matrix is:

$$\begin{bmatrix} K_1 & -K_1 & 0 \\ -K_1 & K_1 + K_2 & -K_2 \\ 0 & -K_2 & K_2 \end{bmatrix} = \begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix}$$

The equation system becomes at first:

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 5{,}000 \end{bmatrix}$$

Installing the boundary conditions: $U_1 = 0$ according to method 1:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 5{,}000 \end{bmatrix}$$

The solution of this equation system is:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0{,}1214 \\ 0{,}3641 \end{bmatrix}$$

Now we do a back-calculation of the internal forces according to method 2 to receive the rod forces:

element 1

$$\begin{bmatrix} 41{,}200 & -41{,}200 \\ -41{,}200 & 41{,}200 \end{bmatrix} \begin{bmatrix} 0 \\ 0.1214 \end{bmatrix} = \begin{bmatrix} -5{,}000 \\ +5{,}000 \end{bmatrix}$$

Figure 2-8: Forces in the nodes of rod 1

<u>element 2</u>

$$\begin{bmatrix} 20{,}600 & -20{,}600 \\ -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} 0.1214 \\ 0.3641 \end{bmatrix} = \begin{bmatrix} -5{,}000 \\ +5{,}000 \end{bmatrix}$$

Figure 2-9: Forces in the nodes of rod 2

In the example the force F_3 was given. Now we have to set a defined displacement, instead of the force. This is what the equation system looks like at first:

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

We set the external forces:

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

If we do not apply any external forces, all of them are 0.

Now we have to set a displacement $U_3 = 0.3641$ mm. Since it is different from 0, we call it *inhomogeneous boundary condition*:

METHOD 3: APPLYING INHOMOGENEOUS BOUNDARY CONDITIONS

The inhomogeneous boundary condition has the value C_j and is valid in the degree of freedom j.

m3.1: Substract the column vector, which is the product of C_j and the column j from $\boldsymbol{K}$, from the right hand side $\boldsymbol{F}$.

m3.2: Apply method 1.

m3.3: Substitute F_j with C_j.

Trying this out immediately shows:

<u>Step m3.1:</u>

$U_3 = 0.3641 = C_j$ i.e. $j = 3$

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & -20{,}600 \\ 0 & -20{,}600 & 20{,}600 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 - 0.3641 \cdot 0 \\ 0 - 0.3641 \cdot (-20{,}600) \\ 0 - 0.3641 \cdot 20{,}600 \end{bmatrix} = \begin{bmatrix} 0 \\ +7{,}500.46 \\ -7{,}500.46 \end{bmatrix}$$

Step m3.2:

Applying method 1, i.e. zero in row 3 and column 3 in $\boldsymbol{K}$, the diagonal element K_{33} is 1, F_3 is 0:

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ +7{,}500.46 \\ 0 \end{bmatrix}$$

U_3 would be 0. Unambiguously wrong! Now we have to set F_3 to C_3=U_3 = 0.3641.

Step m3.3:

$$\begin{bmatrix} 41{,}200 & -41{,}200 & 0 \\ -41{,}200 & 61{,}800 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 7{,}500.46 \\ 0.3641 \end{bmatrix}$$

Now we apply the boundary condition U_1 = 0, i.e. the left fixed bearing, according to method 1:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 61{,}800 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 7{,}500.46 \\ 0.3641 \end{bmatrix}$$

The solution of the equation system is:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1214 \\ 0.3641 \end{bmatrix}$$ ... and that is right!

Arithmetic example 3

The following shows the entire procedure with a beam in the plane at the example of a support:

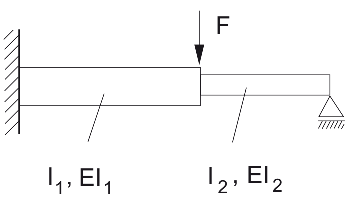


Figure 2-10: Forces in the beam

The system is statically over-determined. However, this does not bother us at all. One of the big advantages of the FEA is that we can calculate arbitrarily statically over-determined systems. Compared to the “hand calculation” with the “0”- and “1”- or “2”-... “n”- system of the engineering mechanics, which becomes costlier with every additional over-determined value, the arithmetic expenditure of FEA does not noticeably increase. Hence, FEA is very suitable to calculate arbitrarily statically over-determined frameworks of trusses and beams or continuous beams.

However, when calculating with the classical engineering mechanics we have to show initiative already with this simply statically over-determined system: Either you take away the right support and compensate the appearing displacement w with a force X so that the displacement in the right support becomes 0 (see Figure 2-11), or you take away the moment restraint in the left support and compensate the now appearing angle of twist ψ with the moment $\widehat{X}$ (see Figure 2-12). By using the FEA you do not need to carry out any of these considerations anymore.

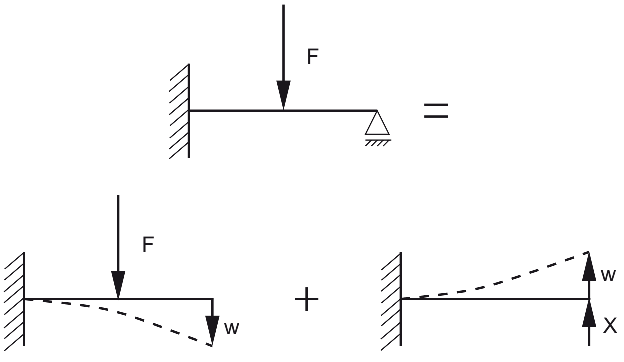


Figure 2-11: Statically over-determined value X as a force

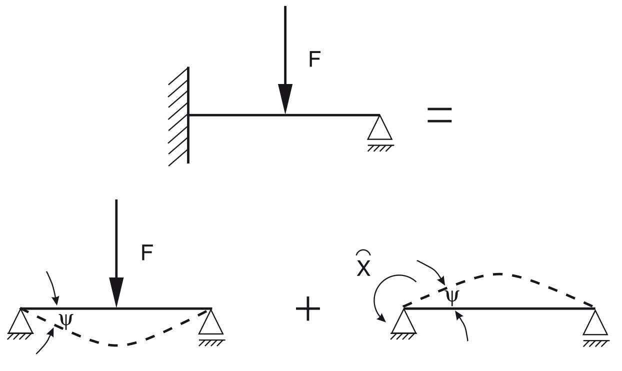


Figure 2-12: Statically over-determined value $\widehat{X}$ as a moment

With this we need a horizontally lying beam in the plane:

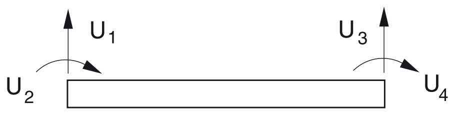


Figure 2-13: The deflections in the beam

Its element stiffness matrix (at this moment we extract it simply from the literature without inquiries) is as follows:

$$EI \cdot \begin{bmatrix} \frac{12}{\ell^3} & \frac{-6}{\ell^2} & \frac{-12}{\ell^3} & \frac{-6}{\ell^2} \\ \frac{-6}{\ell^2} & \frac{4}{\ell} & \frac{6}{\ell^2} & \frac{2}{\ell} \\ \frac{-12}{\ell^3} & \frac{6}{\ell^2} & \frac{12}{\ell^3} & \frac{6}{\ell^2} \\ \frac{-6}{\ell^2} & \frac{2}{\ell} & \frac{6}{\ell^2} & \frac{4}{\ell} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}$$

or

$$\boldsymbol{K}\,\boldsymbol{U} = \boldsymbol{F}$$

or

$$\sum_j K_{ij}\, U_j = F_i$$

We find such element stiffness matrices in some of the literature listed in the bibliography */1–7/* or in chapter 4.

Some authors display the circumstances as follows:

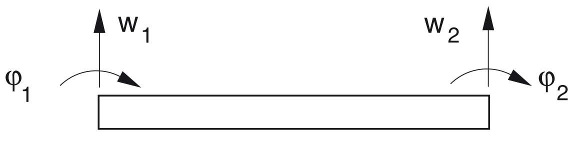


Figure 2-14: Alternative representation of the deflections

$$EI \cdot \begin{bmatrix} \frac{12}{\ell^3} & \frac{-6}{\ell^2} & \frac{-12}{\ell^3} & \frac{-6}{\ell^2} \\ \frac{-6}{\ell^2} & \frac{4}{\ell} & \frac{6}{\ell^2} & \frac{2}{\ell} \\ \frac{-12}{\ell^3} & \frac{6}{\ell^2} & \frac{12}{\ell^3} & \frac{6}{\ell^2} \\ \frac{-6}{\ell^2} & \frac{2}{\ell} & \frac{6}{\ell^2} & \frac{4}{\ell} \end{bmatrix} \begin{bmatrix} w_1 \\ \varphi_1 \\ w_2 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \end{bmatrix}$$

Superficially we can see the displacements w, twists φ, forces F and moments M are affecting the beam, but the desired schematic treatment is complicated. Especially for the representation

$$\sum_j K_{ij}\, U_j = F_i$$

it is absolutely not suitable. We particularly need the index form of the matrix notation for the programming.