

Andres Lahe

THE EST METHOD

Structural analysis

Compatibility equations of displacement

Joint equilibrium equation

This book proposes a method, devised by the author, of solving boundary value problems for the structural analysis of frames, beams, and trusses. Differential equations are considered here together with a set of boundary conditions. Compatibility equations for joint displacements and equilibrium equations for member end forces at joints serve as boundary conditions. The EST method programs written in GNU Octave language assemble and solve sparse systems of equations with unknown member end displacements and forces. The analysis technique is illustrated with numerous examples accompanied with GNU Octave programs.



Andres Lahe (born March 30, 1938 in Kõnnu Parish, Harju County, Estonia), since 2005 Professor Emeritus at Tallinn University of Technology. 1956–1959 Tartu University. 1964 Diploma in Engineering (MEng) from Tallinn University of Technology (TUT). 1965–1970 external student at Leningrad State University, Faculty of Mathematics and Mechanics. 1965–1970 postgraduate studies at the Institute of Cybernetics, the Estonian Academy of Sciences. 1973 Cand. Sc. (PhD) in physics and mathematics (thesis on nonlinear wave propagation in plates and shells). 1959–2004 lab technician, engineer, senior lecturer, professor at TUT.

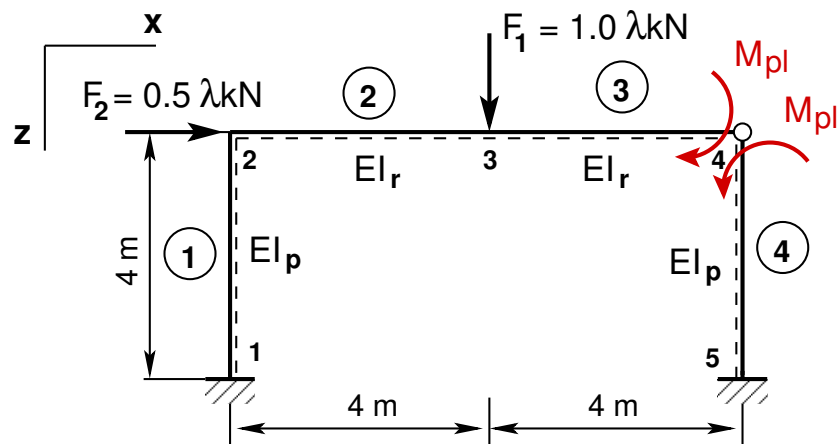




Andres Lahe

The EST Method

Structural analysis



Published by [Tallinn University of Technology Press](#)

Copyright: [Andres Lahe](#), 2014

ISBN 978-9949-23-677-0 (publication)

ISBN 978-9949-23-678-7 (PDF)

Printed in Estonia by [Tallinna Raamatutrükikoda](#)



This book is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.



Program excerpts in the book are subject to the terms of the GNU GPL programs License 2.0 or later. To view a copy of the GNU General Public License license, visit <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html> or write to the Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

The development of sparse matrix algorithms has led to changes in computational structural mechanics and in the corresponding methods of analysis. The EST method is a most general method of analysis for framed structures: beams, trusses, and frames. In this method, compatibility equations for displacements and equilibrium equations for the member end forces at joints are incorporated. This book considers the solving of boundary value problems of frames, beams and trusses with the EST method. The boundary value problem (differential equations together with a set of boundary conditions) is well posed in the method. Solutions to differential equations are represented with the initial parameter method (see also the universal equation of elastic curve of a beam [KL09] and the transfer matrix method). Kinematic and static boundary conditions are composed by a computer program. A frame member has 6 initial parameters and 6 end variables (3 displacements and 3 forces). There are $N \times 12 + N_c$ unknowns in the system of sparse equations for a frame (N is the number of elements, N_c – that of support reactions). To assemble and solve boundary problem equations, the following steps are to be made:

1. writing the basic equations of a frame,
2. writing the compatibility equations of displacements at joints,
3. writing the joint equilibrium equations,
4. writing the side conditions,
5. writing the restrictions on support displacements,
6. solving the compiled system of sparse equations,
7. output: initial parameter vectors for element displacements and forces; support reactions.

Round-off errors are reduced by scaling (multiplying) the displacements and rotations by the basic stiffness (scaling multiplier). After solving a system of equations for boundary value problem, the displacements and rotations found are divided by the basic stiffness.

In Chapter 1 of the book, a brief introduction to the fundamental relations of a frame element is given. Chapter 2 deals with the derivations of the EST method equations. The GNU Octave function `LaheFrameDFIm.m` of assembling and solving the boundary problem has been tested¹ with different input data (frames) shown in Chapter 3. The

¹. [./EST_method_examples.pdf](#). Can be found on the CD attached to the book.

procedure of computing statically determinate frames with or without the displacements and rotations (with or without the compatibility equations of the displacements at joints) is considered in Chapter 4. Second-order structural analysis and the EST method for a second-order theory are treated in Chapters 5 and 6. Chapters 7 and 8 deal with the incremental loading method of plastic analysis for framed structures. A full plastic moment at a plastic hinge is described by the side conditions of the EST method.

The book deals with equations in sparse matrix form. Appendix A contains a summary of sparse matrices. Descriptions and links to the GNU Octave programs of the EST method are presented in Appendix D.

The motivation to compose the EST method has come from the books [PL63], [Krä90], [Krä91a], [Krä91b], [KW90], and [Krä91c]. The method outlined in [Lah97a], [Lah97b], and [Lah98a] differs from the transfer matrix method [PL63], [LT80] and boundary element method [Str89], [Har87], [BW80].

I am obliged to Aime-Rutt Hall for correcting my English and Tiia Eikholm for offering the cover design.

This book is dedicated to the memory of my wife [Lilja](#).

Andres Lahe

Table of Contents

I	First order structural analysis	15
1	Introduction	17
1.1	The frame element fundamental relations	17
1.2	The basic equations of beam/frame	22
1.2.1	The Euler Bernoulli beam equation	22
1.2.2	The basic equations of frame element	25
1.2.3	The basic system of equations of frame element	27
1.3	The system equations of frame	31
2	The equations of the EST method	35
2.1	The basic equations of system	35
2.2	The compatibility equations of the displacements	36
2.2.1	The compatibility conditions at the joints of a beam	41
2.3	The joint equilibrium equations	42
2.3.1	The equilibrium equations at the joints of a beam	47
2.4	The side conditions (Ger. die Nebenbedingungen)	48
2.5	The restrictions on the supports displacements	49
2.6	The initial parameter vector for element	51
2.7	The element displacements and forces	52
2.8	The truss element	54
3	Statically indeterminate problems	57
3.1	Illustrative frame problem 1	57
3.2	Illustrative frame problem 2	65
3.3	Illustrative frame problem 3	73
3.4	Illustrative continuous beam problem 1	82
3.5	Illustrative truss problem 1	96
4	The EST method. Statically determinate problems	103
4.1	Illustrative frame problem 1	103
4.2	Illustrative beam problem 1	107
4.3	Illustrative truss problem 1	110

II	Second order structural analysis	111
5	Second order structural analysis	113
5.1	Introduction	113
5.2	The governing equation of a beam-column	113
5.3	The solutions of a beam-column governing equation	119
5.3.1	The set of solutions to homogeneous differential equation	119
5.3.2	Transfer matrix for a beam element with an axial force	122
5.3.3	The particular solution of non-homogeneous differential equation	124
5.3.4	Transformation of a transfer matrix	130
6	The EST method for second order theory	135
6.1	The system equations of frame	135
6.2	The boundary conditions of system	137
6.3	Illustrative frame problem 1	138
6.4	Illustrative frame problem 2	141
III	Plasticity and limit design	143
7	Plasticity and limit design	145
7.1	Introduction	145
7.2	Piirkoormuse määramise meetodid	145
8	The EST method. Limit design probleme	149
8.1	Illustrative probleme	149
IV	Appendices	153
A	Matrices	155
A.1	Sparse Matrices and GNU Octave	155
A.1.1	Introduction to sparse matrices	155
A.1.2	Creating Sparse Matrices	158
A.1.3	The sparse functions	160
A.2	The transformation Matrices	162
B	The energy equations	165
B.1	Works of the internal and external forces.	165
C	Ülekandemaatriksid	169
C.1	Esimest järku teooria ülekandemaatriks	169
C.2	Ülekandemaatriksid teist järku teoorias	171

D Computer Programs	181
D.1 Esimest järku teooria ülekandemaatriksid	181
D.2 Piirkoormuse järgi arvutus	199
D.3 Teist järku teooria järgi arvutus	200
D.4 Teist järku teooria ülekandemaatriksid	201
Bibliography	207
Complementary Bibliography	212

List of Figures

1.1	Rigid body displacements and rotation	17
1.2	Sign Convention	18
1.3	Right-handed and left-handed coordinates	18
1.4	Moment diagrams for unit force Q and unit moment M	18
1.5	Beam element deformations	19
1.6	Moment diagrams for force Q and moment M	19
1.7	Shear force diagrams for force Q and unit force q	20
1.8	Bending of a frame element (Sign Convention 2)	21
1.9	A differential frame element	22
1.10	External reactions and restraints on displacements	26
1.11	Frame element (Sign Convention 2)	28
1.12	Internal reactions (contact forces) and displacements of joints	29
1.13	Structure of the system of boundary value problem equations	31
1.14	Angles between the vector and the coordinate axes	32
2.1	Transitivity	36
2.2	Compatibility at rigid joint $A3$	36
2.3	Compatibility at hinged joint $A2$	36
2.4	Compatibility at rigid joint $B3$	37
2.5	Compatibility at rigid & pin joint $B32$	37
2.6	Compatibility at pin joint $B2$	38
2.7	Compatibility at rigid joint $T2$	40
2.8	Compatibility at pin joint $T1$	40
2.9	Equilibrium of joint $A3$	41
2.10	Equilibrium of joint $A2$	41
2.11	Equilibrium of joint $B3$	41
2.12	Equilibrium of joint $B2$	42
2.13	Equilibrium of joint $B32$	42
2.14	Equilibrium of support node $A3s$	43
2.15	Equilibrium of support node $A2v$	43
2.16	Equilibrium of support node $A2p$	43
2.17	Equilibrium of joint $T2s$	45
2.18	Equilibrium of joint $T1s$	45
2.19	Cross-sectional directions	51

2.20	Truss element	52
3.1	The two-span frame	57
3.2	Numeration of displacements at frame element ends	60
3.3	Numeration of forces at frame element ends	60
3.4	Bending moment diagram of the two-span frame	62
3.5	Diagrams of the axial force N and shear force Q of the two-span frame	63
3.6	Boundary forces of the two-span frame	63
3.7	Elements of the two-span frame	64
3.8	Sparsity pattern of matrix spA of the two-span frame	64
3.9	Two-span frame EST93	65
3.10	Numeration of displacements and forces of the two-span frame EST93	66
3.11	Internal forces diagrams of the two-span frame EST93	70
3.12	Boundary forces of the two-span frame EST93	71
3.13	Elements of the two-span frame EST93	72
3.14	Sparsity pattern of matrix spA of the two-span frame EST93	72
3.15	The frame EST77	73
3.16	Numeration of displacements and forces of the frame EST77	74
3.17	Internal forces diagrams of the frame EST77	78
3.18	Boundary forces of the frame EST77	80
3.19	Elements of the frame EST77	81
3.20	Sparsity pattern of matrix spA of the frame EST77	81
3.21	The continuous beam EST	82
3.22	Numeration of displacements and forces of the continuous beam EST	83
3.23	Internal forces diagrams of the continuous beam (1st loading)	88
3.24	Internal forces diagrams of the continuous beam (2nd loading)	90
3.25	Internal forces diagrams of the continuous beam (3rd loading)	92
3.26	The continuous beam minimum-maximum bending moments	94
3.27	Elements of the continuous beam EST	95
3.28	Sparsity pattern of matrix spA of the continuous beam EST	95
3.29	The truss EST	96
3.30	A free-body diagram of the truss with joint numbers	97
3.31	Numeration of displacements and forces of the truss EST	99
3.32	Support reactions of the truss EST	100
3.33	Elements of the truss EST	102
3.34	Sparsity pattern of matrix spA of the truss	102
4.1	The three-hinged frame EST1	103
4.2	Numeration of displacements and forces of the frame EST1	104
4.3	Support reactions of the frame EST1	108
4.4	Internal forces diagrams of the three-hinged frame EST1	108
4.5	Elements of the three-hinged frame EST1	109
4.6	Sparsity pattern of matrix spA of the three-hinged frame EST1	109
4.7	The three-span Gerber beam EST	110

4.8	Numeration of displacements and forces of the Gerber beam	113
4.9	Internal forces diagrams of the Gerber beam EST	115
4.10	Elements of the Gerber beam EST	116
4.11	Sparsity pattern of matrix spA of the Gerber beam EST	116
4.12	A polygonal eight-panel truss	117
4.13	Influence line for member 9	120
4.14	Sparsity pattern of matrix spA of the polygonal truss	122
5.1	Normal, shear, axial and transverse shear forces	125
5.2	Deformed beam elements with the axial and normal forces	126
5.3	Components of the normal force	128
5.4	Bending moments of a beam-column	129
5.5	Axial and normal forces	142
6.1	The frame EST1	149
6.2	Numeration of displacements and forces of the frame EST1	150
6.3	Internal forces diagrams of the frame EST1	154
6.4	Boundary forces of the frame EST1	155
6.5	Elements of the frame EST1	156
6.6	Sparsity pattern of matrix spA of the frame EST1	156
6.7	The two-span frame EST2	157
6.8	Displacement and force numbers of the two-span frame EST2	158
6.9	Horizontal displacements diagram of the frame EST2	161
6.10	Internal forces diagrams of the two-span frame EST2	162
6.11	Boundary forces of the two-span frame EST2	164
6.12	Elements of the frame EST2	165
6.13	Sparsity pattern of matrix spA of the frame EST2	165
7.1	Stress-strain curve	169
7.2	Rectangular cross section	170
7.3	Shape factor α_{pl} for different sections	171
7.4	Cross section and stresses	171
7.5	Work done in a plastic hinge	172
7.6	Upper and lower bound values	173
8.1	The frame ESTsn3	177
8.2	Bending moment diagrams of an $n = 3$ times indeterminate frame	179
8.3	Bending moment diagrams of an $n = 2$ times indeterminate frame	179
8.4	Bending moment diagrams of an $n = 1$ times indeterminate frame	179
8.5	Bending moment diagrams of a statically determined frame ($n = 0$)	180
8.6	Load-deflection relation for the frame ESTsn3	181
8.7	Load cases of the $n = 3$ times indeterminate frame ESTsn3	181
8.8	Numeration of displacements and forces of the frame ESTsn3	187
8.9	Load cases of the $n = 2$ times indeterminate frame ESTsn2	188
8.10	Load cases of the $n = 1$ times indeterminate frame ESTsn1	194

8.11	Load cases of the $n = 0$ times indeterminate frame ESTsn0	201
A.1	Sparsity pattern of matrix spA	211
A.2	Assembly sequence of a Gerber beam	212
A.3	Coordinate transformation	218
A.4	Direction cosines of an element	219
B.1	Bar member a–b	222

List of Tables

3.1	Comparison of the internal forces of the truss	101
6.1	Comparison of the internal reactions values of the frame EST2	166
C.1	Loading vectors (to be continued in Table C.2)	226
C.2	Loading vectors (continued from Table C.1)	227
C.3	Loading vectors of compression (axial force S and transverse shear force H)	229
C.4	Loading vectors of tension (axial force S and transverse shear force H)	230
C.5	Loading vectors of compression (normal force N and shear force Q)	231
C.6	Loading vectors of tension (normal force N and shear force Q)	232

Part I

First-order structural analysis

1. Introduction

1.1 Fundamental relations of a frame element

The motion of a frame element is composed of rigid body translation, rigid body rotation, and strain producing deformations (see [PW94] p. 171). Let us now apply a local coordinate system (x, z) to the frame element shown in Fig. 1.1. (X, Z) is the global coordinate system. We consider the right-handed coordinates shown in Fig. 1.3 and Sign Convention 2 in Fig. 1.2.

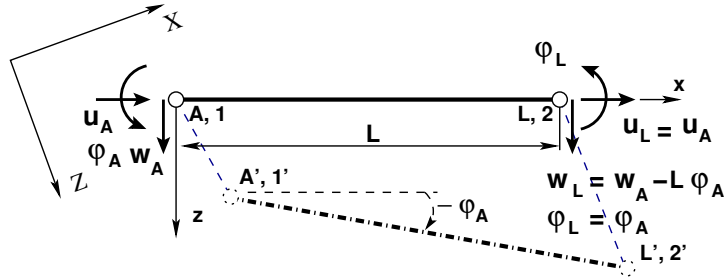


Figure 1.1. Rigid body displacements and rotation

We determine first the rigid body translation and rigid body rotation at the ends of the frame element shown in Fig.1.1:

$$\begin{bmatrix} u_L \\ w_L \\ \varphi_L \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -L \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_A \\ w_A \\ \varphi_A \end{bmatrix} \quad (1.1)$$

where

u_A, u_L – axial displacements at the beginning and at the end of the element, respectively;
 w_A, w_L – transverse displacements at the beginning and at the end of the element, respectively;

φ_A, φ_L – rotation at the beginning and at the end of the element, respectively.

The rigid body displacements and rotations of the element shown in Fig. 1.1 can be large.

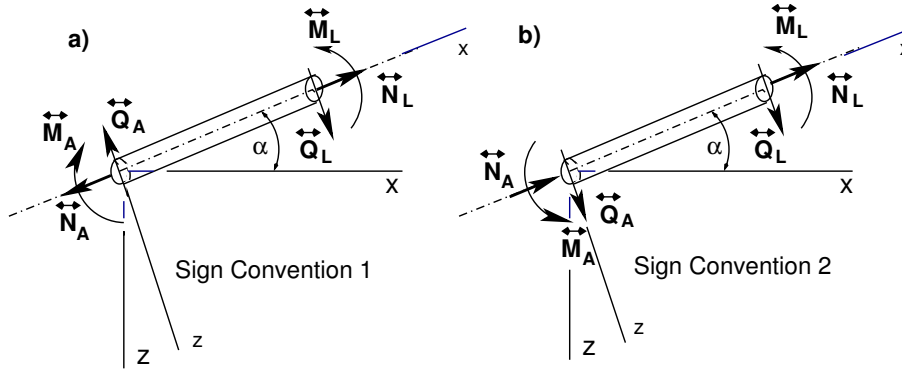
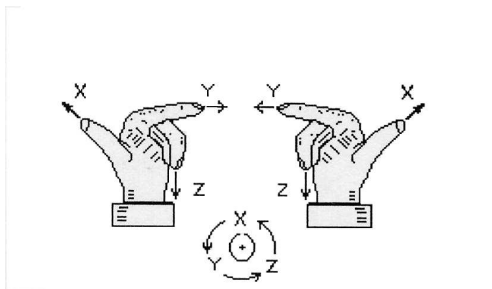


Figure 1.2. Sign Convention

The basic coordinate system (x^*, z^*) in Fig. 1.5 rotates and translates together with the element [Suz00], [YSK08]. With respect to this basic coordinate system, only local strain producing deformations remain (see [PW94] p. 171).



The finger points in the positive direction of the respective axis. Each arrow is labeled by an axis letter in the sequence x, y, z or $1, 2, 3$.

Figure 1.3. Right-handed and left-handed coordinates

The deformations (internal displacements [KHMW10], [KHMW05]) Δw and $\Delta\Theta$ in Fig. 1.5 can be determined by the unit load method using the material law relationships [PW94].

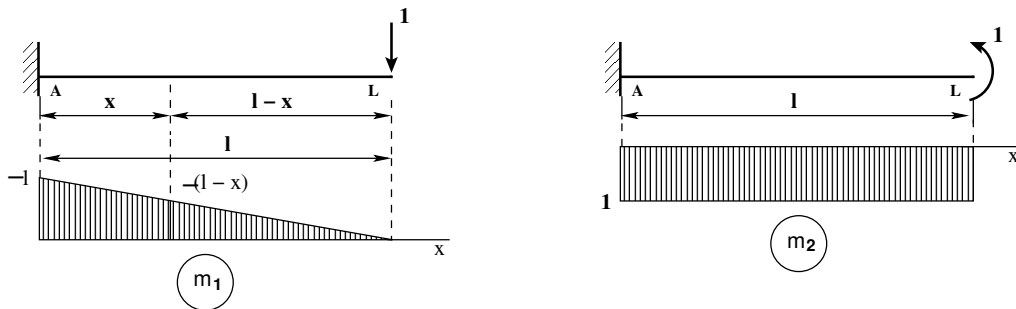


Figure 1.4. Moment diagrams for unit force Q and unit moment M

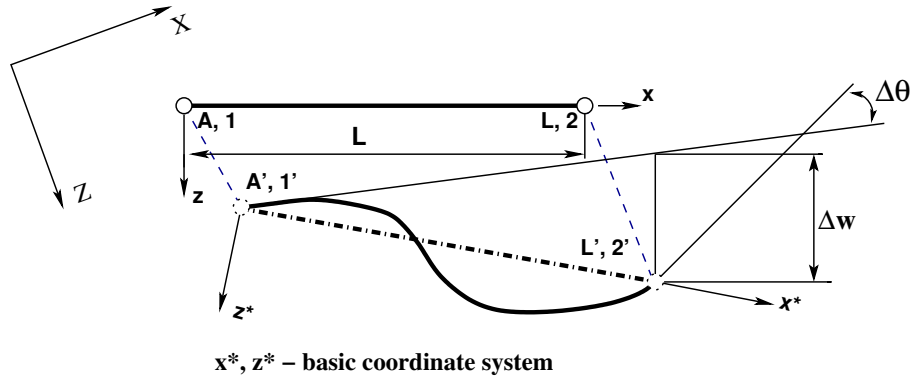


Figure 1.5. Beam element deformations

We proceed decomposing the deformations Δw and $\Delta\Theta$ into the effects of the shear force Q and bending moment M :

$$\begin{bmatrix} \Delta w \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} \Delta w_Q + \Delta w_M \\ \Delta\Theta_Q + \Delta\Theta_M \end{bmatrix} \quad (1.2)$$

Next, we shall consider the moment diagrams in Figs. 1.4, 1.6, and 1.7. When applying the unit load method, it follows that

$$\begin{aligned} \Delta w_Q &= \int_0^l \frac{M_Q m_1}{EI} ds + \int_0^l \frac{Q_Q q_1}{GA_{red}} ds = \frac{l^3 Q_L}{3EI} + \frac{l Q_L}{GA_{red}} \\ \Delta\Theta_Q &= \int_0^l \frac{M_Q m_2}{EI} ds = -\frac{l^2 Q_L}{2EI} \end{aligned} \quad (1.3)$$

where

EI – bending stiffness of the element,

GA_{red} – shear stiffness of the element.

Now we consider again the moment diagrams in Figs. 1.4 and 1.6, and apply the unit load method for finding Δw_M and $\Delta\Theta_M$:

$$\Delta w_M = \int_0^l \frac{M_M m_1}{EI} ds = -\frac{l^2 M_L}{2EI}$$

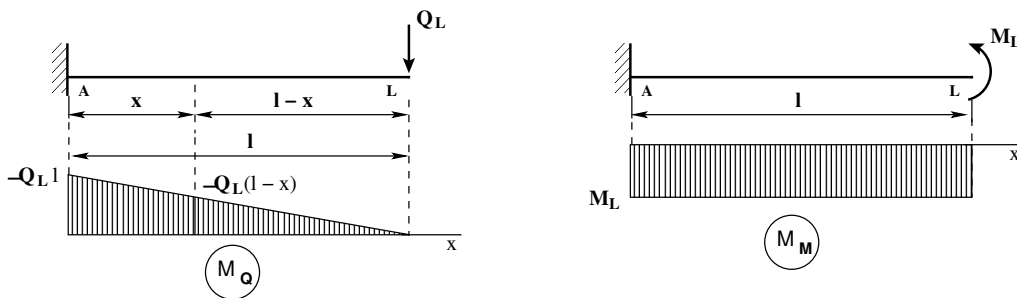


Figure 1.6. Moment diagrams for force Q and moment M

$$\Delta\Theta_M = \int_0^l \frac{M_M m_2}{EI} ds = \frac{lM_L}{EI} \quad (1.4)$$

Substituting Eqs. (1.3), (1.4) into Eq. (1.2), we have

$$\begin{bmatrix} \Delta w \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} \Delta w_Q + \Delta w_M \\ \Delta\Theta_Q + \Delta\Theta_M \end{bmatrix} = \begin{bmatrix} \frac{l^3}{3EI} + \frac{l}{GA_{red}} & -\frac{l^2}{2EI} \\ -\frac{l^2}{2EI} & \frac{l}{EI} \end{bmatrix} \begin{bmatrix} Q_L \\ M_L \end{bmatrix} \quad (1.5)$$

The axial deformation of the element

$$\Delta u_L = \frac{lN_L}{EA} \quad (1.6)$$

where

N_L – axial force at the element end,

l – length of the element,

EA – axial stiffness of the element.

All deformations of the frame element arranged in matrix form:

$$\begin{bmatrix} \Delta u_L \\ \Delta w \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} \frac{l}{EA} & 0 & 0 \\ 0 & \frac{l^3}{3EI} + \frac{l}{GA_{red}} & -\frac{l^2}{2EI} \\ 0 & -\frac{l^2}{2EI} & \frac{l}{EI} \end{bmatrix} \begin{bmatrix} N_L \\ Q_L \\ M_L \end{bmatrix} \quad (1.7)$$

Now we consider a frame element (shown in Fig. 1.8) with no loads applied between the ends. Equilibrium equations of the frame element are written as

$$\begin{bmatrix} N_L \\ Q_L \\ M_L \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & -l & -1 \end{bmatrix} \begin{bmatrix} N_A \\ Q_A \\ M_A \end{bmatrix} \quad (1.8)$$

Using Eq. (1.8), we can rewrite equations (1.7) for all deformations of the frame element in the form

$$\begin{bmatrix} \Delta u_L \\ \Delta w \\ \Delta\Theta \end{bmatrix} = \begin{bmatrix} -\frac{l}{EA} & 0 & 0 \\ 0 & \frac{l^3}{6EI} - \frac{l}{GA_{red}} & \frac{l^2}{2EI} \\ 0 & -\frac{l^2}{2EI} & -\frac{l}{EI} \end{bmatrix} \begin{bmatrix} N_A \\ Q_A \\ M_A \end{bmatrix} \quad (1.9)$$

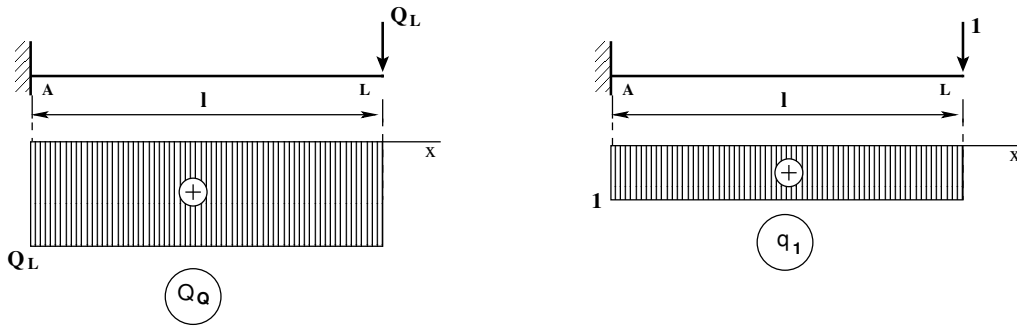


Figure 1.7. Shear force diagrams for force Q and unit force q

We shall now assemble Eqs. (1.1), (1.8), (1.9) into a matrix of basic equations

$$\mathbf{Z}_L = \mathbf{U} \cdot \mathbf{Z}_A \tag{1.10}$$

where $\mathbf{Z}_L, \mathbf{Z}_A$ are the vectors of displacements and forces at the end and at the beginning of the element, respectively,

$$\mathbf{Z}_L = \begin{bmatrix} u_L \\ w_L \\ \varphi_L \\ \dots \\ N_L \\ Q_L \\ M_L \end{bmatrix}, \quad \mathbf{Z}_A = \begin{bmatrix} u_A \\ w_A \\ \varphi_A \\ \dots \\ N_A \\ Q_A \\ M_A \end{bmatrix} \tag{1.11}$$

and \mathbf{U} is the transfer matrix (Sign Convention 2):

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & \vdots & -\frac{l}{EA} & 0 & 0 \\ 0 & 1 & -l & \vdots & 0 & \frac{l^3}{6EI_y} - \frac{l}{GA_{red}} & \frac{l^2}{2EI_y} \\ 0 & 0 & 1 & \vdots & 0 & -\frac{l^2}{2EI_y} & -\frac{l}{EI_y} \\ \dots & \dots & \dots & \vdots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & -1 & 0 & 0 \\ 0 & 0 & 0 & \vdots & 0 & -1 & 0 \\ 0 & 0 & 0 & \vdots & 0 & -l & -1 \end{bmatrix} \tag{1.12}$$

This matrix of the basic equations has the following structure (cf. [PW94]):

$$\mathbf{U} = \begin{bmatrix} \textit{Rigid body} & \vdots & \textit{Internal} \\ \textit{displacements} & \vdots & \textit{displacements} \\ \dots & \vdots & \dots \\ & \vdots & \textit{Equilibrium} \\ & \vdots & \\ & \vdots & \end{bmatrix} \tag{1.13}$$

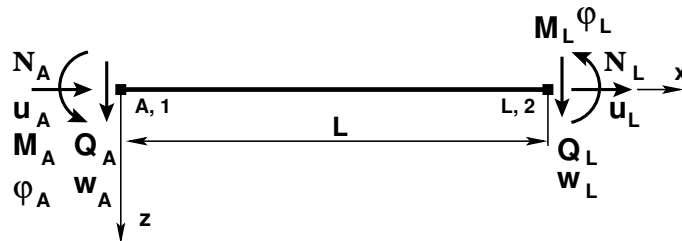


Figure 1.8. Bending of a frame element (Sign Convention 2)

1.2 Basic equations for a beam/frame

There is another possibility for deriving the transfer matrix of Eq. (1.12) for a beam element.

1.2.1 The Euler-Bernoulli beam equation

Given below are the Euler-Bernoulli beam governing differential equations:

$$\begin{aligned} \frac{dw}{dx} &= -\varphi_y \\ \frac{d^2w}{dx^2} &= -\frac{d\varphi_y}{dx} = -\frac{1}{EI_y}M_y \\ \frac{d^3w}{dx^3} &= -\frac{1}{EI_y}\frac{dM_y}{dx} = -\frac{1}{EI_y}Q_z \\ \frac{d^4w}{dx^4} &= -\frac{1}{EI_y}\frac{dQ_z}{dx} = \frac{1}{EI_y}q_z \end{aligned} \quad (1.14)$$

Figure 1.9. A differential frame element

where E is the elastic modulus and I_y is the area moment of inertia.

The last formula of Eq. (1.14) is a non-homogeneous differential equation of 4th order ($d^4w/dx^4 = q_z/EI_y$). We are looking for the general solution of the non-homogeneous differential equation in the form

$$w = w_0w_1 + w'_0w_2 + w''_0w_3 + w'''_0w_4 + w_e(x) \quad (1.15)$$

where

w_0 , w'_0 , w''_0 , and w'''_0 are the values of the sought-for function at $x = x_0$; w_1 , w_2 , w_3 , and w_4 are a normed fundamental set of solutions to the associated homogeneous differential equation; $w_e(x)$ is the particular solution of the non-homogeneous differential equation.

Next we consider a set for fundamental solutions for the associated homogeneous differential equation:

$$w_1 = 1, \quad w_2 = x, \quad w_3 = \frac{x^2}{2}, \quad w_4 = \frac{x^3}{6} \quad (1.16)$$

The Wronskian¹ W of this set of solutions

$$W(x) = \begin{vmatrix} 1 & x & x^2/2 & x^3/6 \\ 0 & 1 & x & x^2/2 \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (1.17)$$

¹The Wronskian is a determinant formulated by the Polish mathematician and philosopher Józef Maria Wroński (1776–1853).

and at $x = 0$

$$W(x=0) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1 \quad (1.18)$$

Thus, the last equation is unit Wronskian and the set of fundamental solutions of Eq. (1.16) is a normed fundamental set of solutions.

By Sign Convention 1, the initial parameters of Eq. (1.15) are

$$w_0 = w_0, \quad w'_0 = -\varphi_0, \quad w''_0 = -\frac{M_0}{EI}, \quad w'''_0 = -\frac{Q_0}{EI} \quad (1.19)$$

and by Sign Convention 2

$$w_0 = w_0, \quad w'_0 = -\varphi_0, \quad w''_0 = \frac{M_0}{EI}, \quad w'''_0 = \frac{Q_0}{EI} \quad (1.20)$$

Substituting Eqs. (1.16) and (1.19) into Eq. (1.15), we get the solution for the homogeneous differential equation (Sign Convention 1):

$$w = \begin{bmatrix} 1 & -x & -x^2/2EI & -x^3/6EI \end{bmatrix} \begin{bmatrix} w_A \\ \varphi_A \\ M_A \\ Q_A \end{bmatrix}; \quad \text{here} \quad \begin{bmatrix} w_A \\ \varphi_A \\ M_A \\ Q_A \end{bmatrix} = \begin{bmatrix} w_0 \\ \varphi_0 \\ M_0 \\ Q_0 \end{bmatrix} \quad (1.21)$$

From Eqs. (1.16) and (1.20) (Sign Convention 2),

$$w = \begin{bmatrix} 1 & -x & x^2/2EI & x^3/6EI \end{bmatrix} \begin{bmatrix} w_A \\ \varphi_A \\ M_A \\ Q_A \end{bmatrix}; \quad \text{here} \quad \begin{bmatrix} w_A \\ \varphi_A \\ M_A \\ Q_A \end{bmatrix} = \begin{bmatrix} w_0 \\ \varphi_0 \\ M_0 \\ Q_0 \end{bmatrix} \quad (1.22)$$

The general non-homogeneous differential equation of the Euler-Bernoulli beam subjected to an external load and equivalent generalized loads is

$$d^4w/dx^4 = q_z/EI_y + \mathcal{F}_z\delta(t-a)/EI_y + \mathcal{M}_x\delta'(t-a)/EI_y \quad (1.23)$$

where

$\mathcal{F}_z\delta(x-a_F)/EI_y$ is the equivalent distributed force q_F of a concentrated force of magnitude \mathcal{F}_z ,

$\mathcal{M}_x\delta'(x-a_M)/EI_y$ is the equivalent distributed force q_M of a concentrated moment of magnitude \mathcal{M}_x ,

$\delta(x-a)$ is the Dirac delta function.

The particular solution $w_e(x)$ of Eq. (1.15) we are looking for is given by the Cauchy formula

$$w_e(x) = \int_{x_0}^x K(x,t) f(t) dt \quad (1.24)$$

or, to be more precise,

$$w_e(x) = \int_{x_0}^x K_4(x, t) f_4(t) dt + \int_{x_0}^x K_3(x, t) f_3(t) dt + \int_{x_0}^x K_2(x, t) f_2(t) dt \quad (1.25)$$

where, using Eq. (1.16),

$$K_4(x, t) = w_4(x - t) = \frac{(x - t)^3}{6} \quad (1.26)$$

$$K_3(x, t) = w_3(x - t) = \frac{(x - t)^2}{2} \quad (1.27)$$

$$K_2(x, t) = w_2(x - t) = (x - t) \quad (1.28)$$

and the load function $f(t)$ is

$$f_4(t) = q(t)/EI, \quad f_3(t) = \mathcal{F}_z/EI_y, \quad f_2(t) = \mathcal{M}_x/EI_y \quad (1.29)$$

The particular solution $w_e(x)$ at a constant load q_z

$$w_e(x) = -\frac{q_z}{EI} \int_{a_q}^x \frac{(x - t)^3}{6} d(x - t) = -\frac{q_z}{EI} \frac{(x - t)^4}{24} \Big|_{a_q}^x = \frac{q_z}{EI} \frac{(x - a_q)_+^4}{24} \quad (1.30)$$

where $(x - a)_+$ is the Heaviside step function:

$$(x - a)_+ = \begin{cases} 0, & \text{if } (x - a) < 0 \\ x - a, & \text{if } (x - a) \geq 0 \end{cases} \quad (1.31)$$

In case of the point load \mathcal{F}_z and moment \mathcal{M}_y , the functions $f(t)$, $K(x, t)$ in the Cauchy formula (1.24) are respectively

$$f_3(t) = \frac{\mathcal{F}_z}{EI}, \quad f_2(t) = \frac{\mathcal{M}_y}{EI} \quad (1.32)$$

$$K_3(x, t) = w_3(x - t) = \frac{(x - t)^2}{2} \quad (1.33)$$

$$K_2(x, t) = w_2(x - t) = (x - t) \quad (1.34)$$

The particular solution $w_e(x)$ in case of the point load \mathcal{F}_z and moment \mathcal{M}_y is respectively

$$w_e(x) = -\frac{\mathcal{F}_z}{EI} \int_{a_F}^x \frac{(x - t)^2}{2} d(x - t) = -\frac{\mathcal{F}_z}{EI} \frac{(x - t)^3}{6} \Big|_{a_F}^x = \frac{\mathcal{F}_z}{EI} \frac{(x - a_F)_+^3}{6} \quad (1.35)$$

$$w_e(x) = -\frac{\mathcal{M}_y}{EI} \int_{a_M}^x (x - t) d(x - t) = -\frac{\mathcal{M}_y}{EI} \frac{(x - t)^2}{2} \Big|_{a_M}^x = \frac{\mathcal{M}_y}{EI} \frac{(x - a_M)_+^2}{2} \quad (1.36)$$

The general solution of the non-homogeneous differential equation

$$d^4w/dx^4 = q_z/EI_y + \mathcal{F}_z\delta(t - a)/EI_y + \mathcal{M}_x\delta'(t - a)/EI_y \quad (1.37)$$

is the sum of the solutions of the homogeneous differential equation (1.22) (Sign Convention 2) and of the particular solutions of Eqs. (1.30), (1.35), and (1.36):

$$w_x = w_A - \varphi_A \cdot x + Q_A \frac{x^3}{6EI} + M_A \frac{x^2}{2EI} + \frac{q_z}{EI} \frac{(x-a_q)_+^4}{24} + \frac{\mathcal{F}_z}{EI} \frac{(x-a_F)_+^3}{6} + \frac{\mathcal{M}_y}{EI} \frac{(x-a_M)_+^2}{2} \quad (1.38)$$

Let us take the derivatives of displacement from Eq. (1.38) and apply those to the governing differential equation (1.14). We obtain beam governing equations with the transfer matrix

$$\underbrace{\begin{bmatrix} w_x \\ \varphi_x \\ \dots \\ Q_x \\ M_x \end{bmatrix}}_{\mathbf{Z}_x} = \underbrace{\begin{bmatrix} 1 & -x & \vdots & x^3/6EI_y - x/GA_{red} & x^2/2EI_y \\ 0 & 1 & \vdots & -x^2/2EI_y & -x/EI_y \\ \dots & \dots & \vdots & \dots & \dots \\ 0 & 0 & \vdots & -1 & 0 \\ 0 & 0 & \vdots & -x & -1 \end{bmatrix}}_{\mathbf{U}_x} \underbrace{\begin{bmatrix} w_A \\ \varphi_A \\ \dots \\ Q_A \\ M_A \end{bmatrix}}_{\mathbf{Z}_A} + \underbrace{\begin{bmatrix} + & \frac{q_z}{EI} \frac{(x-a_q)_+^4}{24} & + & \frac{\mathcal{F}_z}{EI} \frac{(x-a_F)_+^3}{6} & + & \frac{\mathcal{M}_y}{EI} \frac{(x-a_M)_+^2}{2} \\ - & \frac{q_z}{EI} \frac{(x-a_q)_+^3}{6} & - & \frac{\mathcal{F}_z}{EI} \frac{(x-a_F)_+^2}{2} & - & \frac{\mathcal{M}_y(x-a_M)_+}{EI} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ - & q_z(x-a_q)_+ & - & \mathcal{F}_z & - & 0 \\ - & q_z \frac{(x-a_q)_+^2}{2} & - & \mathcal{F}_z(x-a_F)_+ & - & \mathcal{M}_y \end{bmatrix}}_{\mathring{\mathbf{Z}}} \quad (1.39)$$

where the effect of the shear force $(-x/GA_{red})$ from Eq. (1.3) is added to the deformation.

In symbolic matrix notation, the formulae of Eq. (1.39) can be written as

$$\mathbf{Z}_x = \mathbf{U}_x \mathbf{Z}_A + \mathring{\mathbf{Z}} \quad (1.40)$$

1.2.2 Basic equations for a frame element

The governing differential equation for a truss element is

$$\frac{dN_x}{dx} = -q_x(x) \quad (1.41)$$

or

$$\frac{d}{dx} \left(EA \frac{du}{dx} \right) = -q_x(x) \quad (1.42)$$

	Fixed support	Pin support	Roller support		Free end	Contrived end
Support symbols						
Support conditions: restraints on support displacements	$u = 0$ $F_x = ?$ $F_z = ?$ $M_y = ?$ $w = 0$ $\varphi_y = 0$	$u = 0$ $F_x = ?$ $F_z = ?$ $M_y = 0$ $w = 0$ $\varphi_y = ?$	$u = ?$ $F_x = 0$ $F_z = ?$ $M_y = 0$ $w = 0$ $\varphi_y = ?$	$u = 0$ $F_x = ?$ $F_z = 0$ $M_y = 0$ $w = ?$ $\varphi_y = ?$	$u = ?$ $F_x = 0$ $F_z = 0$ $M_y = 0$ $w = ?$ $\varphi_y = ?$	$u = 0$ $F_x = ?$ $F_z = 0$ $M_y = ?$ $w = ?$ $\varphi_y = 0$
Support displacements	— 0	φ_y 1	u, φ_y 2	w, φ_y 2	u, w, φ_y 3	w 1
Support reactions	 3	 2	 1	 1	0	 2

Figure 1.10. External reactions and restraints on displacements

where $N_x = EAdu/dx$ and EA denotes the axial stiffness.

We now consider the basic equations of a frame element in symbolic matrix notation

$$\mathbf{Z}_x = \mathbf{U}_x \mathbf{Z}_A + \overset{\circ}{\mathbf{Z}} \tag{1.43}$$

where $\mathbf{Z}_x, \mathbf{Z}_A$ are the vectors of displacements and forces at the point with x -coordinate and at the beginning of the element, respectively,

$$\mathbf{Z}_x = \begin{bmatrix} u_x \\ w_x \\ \varphi_x \\ \dots \\ N_x \\ Q_x \\ M_x \end{bmatrix}, \quad \mathbf{Z}_A = \begin{bmatrix} u_A \\ w_A \\ \varphi_A \\ \dots \\ N_A \\ Q_A \\ M_A \end{bmatrix} \quad (1.44)$$

and \mathbf{U}_x is the transfer matrix (Sign Convention 2):

$$\mathbf{U}_x = \begin{bmatrix} 1 & 0 & 0 & \vdots & -i_0 \frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & \vdots & 0 & i_0 \frac{x^3}{6EI_y} - i_0 \frac{x}{GA_{red}} & i_0 \frac{x^2}{2EI_y} \\ 0 & 0 & 1 & \vdots & 0 & -i_0 \frac{x^2}{2EI_y} & -i_0 \frac{x}{EI_y} \\ \dots & \dots & \dots & \vdots & \dots & \dots & \dots \\ 0 & 0 & 0 & \vdots & -1 & 0 & 0 \\ 0 & 0 & 0 & \vdots & 0 & -1 & 0 \\ 0 & 0 & 0 & \vdots & 0 & -x & -1 \end{bmatrix} \quad (1.45)$$

where $i_0 = 1$ is the scaling multiplier for the displacements ($i_0 = EI/l$) and the loading vector ([yzhqz.m](#), [yzfzv.m](#), [yzmyv.m](#)) can be expressed as

$$\overset{\circ}{\mathbf{Z}} = \begin{bmatrix} -i_0 \frac{q_x(x-x_a)_+^2}{2*EA^4} \\ i_0 \frac{q_z(x-x_a)_+}{24EI_y} \\ -i_0 \frac{q_z(x-x_a)_+^3}{6EI_y} \\ -q_x(x-x_a)_+ \\ -q_z(x-x_a)_+ \\ -q_z(x-x_a)_+^2/2 \end{bmatrix} + \begin{bmatrix} -i_0 \frac{F_x(x-x_a)_+}{EA} \\ i_0 \frac{F_z(x-x_a)_+^3}{6EI_y} \\ -i_0 \frac{F_z(x-x_a)_+^2}{2EI_y} \\ -F_x(x-x_a)_+^o \\ -F_z(x-x_a)_+^o \\ -F_z(x-x_a)_+ \end{bmatrix} + \begin{bmatrix} 0 \\ i_0 \frac{M_y(x-x_a)_+^2}{2EI_y} \\ -i_0 \frac{M_y(x-x_a)_+}{EI_y} \\ 0 \\ 0 \\ -M_y(x-x_a)_+^o \end{bmatrix} \quad (1.46)$$

Some other loading vectors are to be found in Tables [C.1](#) and [C.2](#) of Appendix [C](#).

1.2.3 The basic system of equations for a frame element

We consider a frame element with loads applied between the ends (Fig. 1.11).

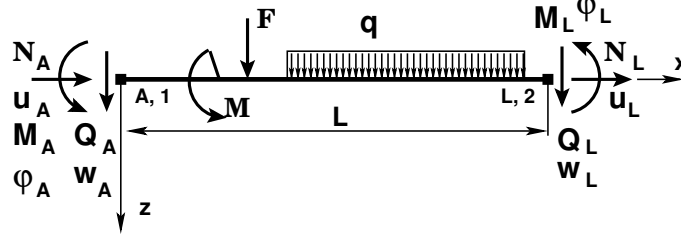


Figure 1.11. Frame element (Sign Convention 2)

Here, we rewrite Eq. (1.40) in the form

$$\mathbf{I}_{6 \times 6} \cdot \mathbf{Z}_L - \mathbf{U}_{x=l} \mathbf{Z}_A = \mathbf{Z}_{x=l}^{\circ} \quad (1.47)$$

or

$$\widehat{\mathbf{I}}\mathbf{U}_{6 \times 12} \cdot \widehat{\mathbf{Z}} = \mathbf{Z}^{\circ} \quad (1.48)$$

where $\widehat{\mathbf{I}}\mathbf{U}$ is the *enlarged transfer matrix* \mathbf{U} :

$$\widehat{\mathbf{I}}\mathbf{U} = \mathbf{I} - \mathbf{U}, \quad (1.49)$$

$\widehat{\mathbf{Z}}$ is the vector of **displacements and forces (DaFs)** of the element:

$$\widehat{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_L \\ \mathbf{Z}_A \end{bmatrix} \quad (1.50)$$

\mathbf{Z}_L , \mathbf{Z}_A are the displacement and force vectors at the end and at the beginning of the element, respectively,

\mathbf{I} is a unit matrix (for the frame element 6×6),

\mathbf{U} is the transfer matrix of Eq. (1.45) at $x = l$ (Sign Convention 2),

\mathbf{Z}° is the loading vector of Eq. (1.46) at $x = l$ (Sign Conventions 1 and 2).

Note that Eq. (1.42) is a second-order equation and requires two boundary conditions, whereas Eq. (1.14) is a fourth-order equation requiring four boundary conditions. These (2+4=6) boundary conditions should be well posed. They are divided into static and kinematic boundary conditions.

The fixed-end displacements and rotation at the end and at the beginning of the element are respectively

$$\mathbf{v}_L = \begin{bmatrix} u_L \\ w_L \\ \varphi_L \end{bmatrix}, \quad \mathbf{v}_A = \begin{bmatrix} u_A \\ w_A \\ \varphi_A \end{bmatrix} \quad (1.51)$$

	Rigid joint	Bending moment hinge	Axial force hinge	Shear force hinge	Free ends
Symbols of hinges					
Internal displacements	$\Delta u = u_p - u_v$ $\Delta w = w_p - w_v$ $\Delta \varphi_y = \varphi_p - \varphi_v$				
Given DaFs	$\Delta u = 0$ $\Delta w = 0$ $\Delta \varphi_y = 0$	$\Delta u = 0$ $\Delta w = 0$ $\Delta \varphi_y = ?$	$\Delta u = ?$ $\Delta w = 0$ $\Delta \varphi_y = 0$	$\Delta u = 0$ $\Delta w = ?$ $\Delta \varphi_y = 0$	$\Delta u = ?$ $\Delta w = ?$ $\Delta \varphi_y = ?$
Unknown DaFs	$N_x = ?$ $Q_z = ?$ $M_y = ?$	$N_x = ?$ $Q_z = ?$ $M_y = 0$	$N_x = 0$ $Q_z = ?$ $M_y = ?$	$N_x = ?$ $Q_z = 0$ $M_y = ?$	$N_x = 0$ $Q_z = 0$ $M_y = 0$
Internal displacements DOF number	0	$\Delta \varphi_y$ 1	Δu 1	Δw 1	$\Delta u, \Delta w, \Delta \varphi_y$ 3
Internal reactions and their number					0

Figure 1.12. Internal reactions^{2,3} (contact forces⁴) and displacements of joints

The fixed-end forces and the moments $\mathbf{s}_L, \mathbf{s}_A$ at the end and at the beginning of the element, respectively, are shown in Eq. (1.52).

At unloaded joints (contacts) of elements, the sum of the fixed-end forces and moments is zero.

²Fixed-end forces and moments at joints are called internal reactions [WP960],

http://engr.bd.psu.edu/rxm61/213/Beams_overview.ppt. Web. 20 August 2013.

³Shear and Moment Diagrams for Frames: ... cut the frame into its component members and find the internal reactions [MPHS].

⁴A contact force is a force that acts at the points of contact between two objects [Rand07].

$$\mathbf{s}_L = \begin{bmatrix} N_L \\ Q_L \\ M_L \end{bmatrix}, \quad \mathbf{s}_A = \begin{bmatrix} N_A \\ Q_A \\ M_A \end{bmatrix} \quad (1.52)$$

Sometimes the fixed-end forces and moments at joints are called internal reactions^{5 6} or joint contact forces⁷ [GN12]. We shall call them *contact forces* and *contact moments*.

A boundary value problem is a differential equation together with a set of boundary conditions. Solutions to the differential equations of a frame element are given with the basic system of equations (1.47). The boundary value problem should be well posed. The kinematic and static boundary conditions [KW90] of a frame element are shown in Figs. (1.10) and (1.12).

1.3 The system equations of a frame

The *total number* of unknowns N_{totl} in this problem is decomposed as follows:

- N_{est} – the number of displacements and forces (see Eq. (1.50)) of the elements,
 - for frames $N_{est} = N_{elems} \times 12$ (N_{elems} – the number of elements),
 - for beams $N_{est} = N_{elems} \times 8$;
- N_{react} – the number of external support reactions of the frame (beam).

$$N_{totl} = N_{est} + N_{react} \quad (1.53)$$

To solve this boundary value problem, it is necessary but not sufficient that the number of linear equations N_{eqs} in a system of equations

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (1.54)$$

be

$$N_{eqs} = N_{totl}, \quad \text{and} \quad \text{rank}(\mathbf{spA}) = N_{totl} \quad (1.55)$$

Both external and internal boundary conditions (Fig. 1.13) of a system (frame/beam) should be well posed.

⁵http://engr.bd.psu.edu/rxm61/213/Beams_overview.ppt. Web. 20 August 2013.

⁶WP 960 Technical Description: ... Two force gauges determine the internal reactions ... [WP960].

⁷A *contact force* is a force that acts at the points of contact between two objects [Rand07].

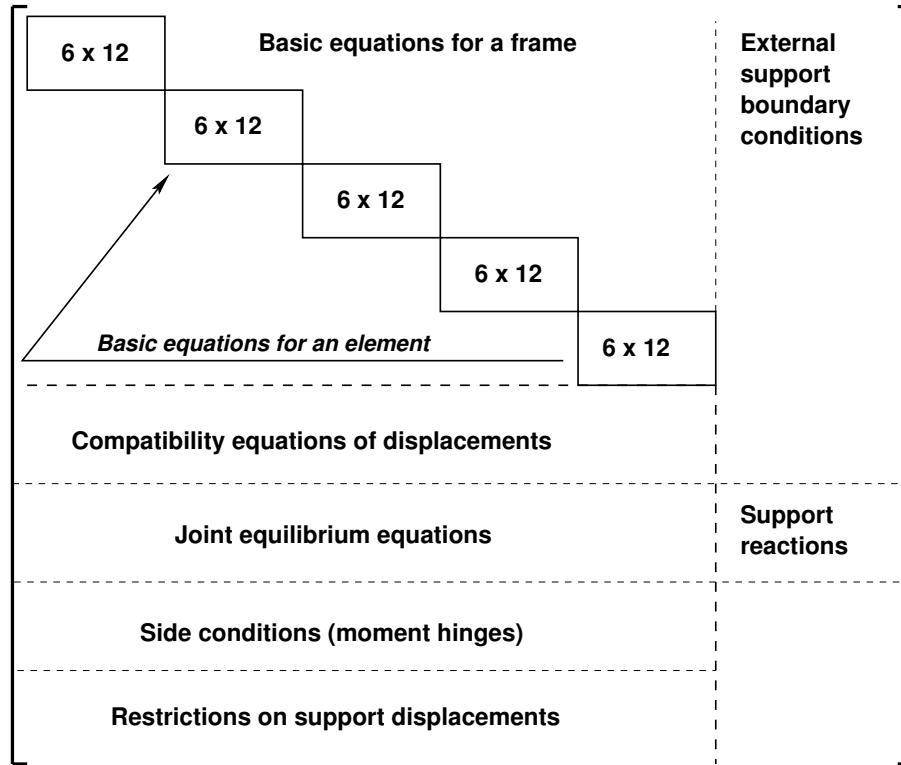


Figure 1.13. Structure of the system of boundary value problem equations

The kinematic and static boundary conditions of the external support shown in Fig. (1.10) and these of the internal support shown in Fig. (1.12) should be well posed.

The kinematic and static boundary conditions of the internal support are divided into

- compatibility equations of the displacements at nodes,
- joint equilibrium equations at nodes,
- side conditions (bending moment, axial and shear force hinges)

It is convenient to use a sparse representation of equations (1.54) (see section A.1).

Numerical difficulties may occur when the transfer matrix manipulation involves differences of large numbers, which can lead to inaccuracies in computations [PW94], [PL63]. In the state vector \mathbf{Z} of equations (1.47) the displacements and rotations are small in comparison with the contact forces and moments. We will scale (multiply) the displacements and rotations by $i_0 = EI_{basic}/l_{basic}$ (as a scaling multiplier for the displacements, basic stiffness is taken).

In the sparse matrix \mathbf{spA} of Eq. (1.54), the basic equations of the system are described in local coordinates. The compatible equations of displacements, joint equilibrium equations, side conditions equations and external support reactions of the system of Eq. (1.54) are described in global coordinates.

The following direction cosines allow us to transform vectors from local to global coordinates. The *direction cosines* of a vector are the cosines of the angles between the vector and the coordinate axes (see Fig. 1.14):

$$\cos \alpha = \frac{\Delta x}{l}, \quad \cos \beta = \frac{\Delta z}{l} \quad (1.56)$$

Here,

$$\Delta x = x_L - x_A, \quad \Delta z = z_L - z_A, \quad l = \sqrt{(\Delta x)^2 + (\Delta z)^2} \quad (1.57)$$

and x_A, z_A, x_L, z_L are the start point and the end point coordinates (Fig. 1.14).

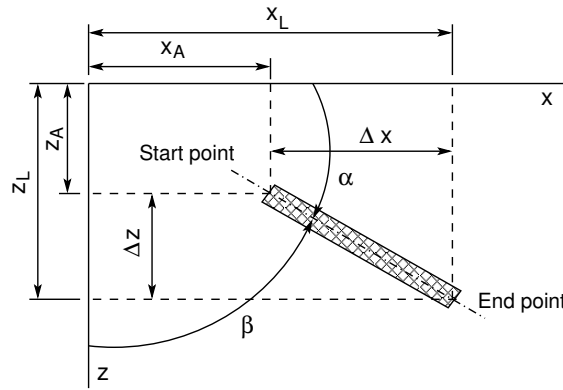


Figure 1.14. Angles between the vector and the coordinate axes

The two-dimensional transformation matrix $\mathbf{T}_{2 \times 2}$ (considered in section A.2) transforms the vector from local to global coordinates.

$$\mathbf{T}_{2 \times 2} = \begin{bmatrix} \cos \alpha & -\cos \beta \\ \cos \beta & \cos \alpha \end{bmatrix} \quad (1.58)$$

The three-dimensional transformation matrix $\mathbf{T}_{3 \times 3}$ (considered in section A.2) transforms the vector from local to global coordinates.

$$\mathbf{T}_{3 \times 3} = \begin{bmatrix} \cos \alpha & -\cos \beta & 0 \\ \cos \beta & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.59)$$

Note that it is possible to insert the transformation matrices of Eqs. (A.25) and (1.59) into the sparse matrix \mathbf{spA} of Eq. (1.54) with the function `spA=spInsert BtoA(spA,M,N,spTi)` (p. 241).

2. Equations of the EST method

The non-symmetric sparse system of linear equations (1.54) has $N_{totl} = N_{est} + N_{react}$ unknowns (see Eq. 1.53):

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (2.1)$$

The structure of the system and the sparsity pattern of the sparse matrix \mathbf{spA} are shown in Figs. 1.13 and 3.14, respectively.

The composition of the EST method equations should be considered with GNU Octave^{1,2} – a high-level interpreted language, primarily intended for numerical calculations [EBH08]. The GNU Octave language is quite similar to Matlab³, so that most programs are easily portable. GNU Octave is distributed under the terms of the GNU General Public License.

2.1 Basic equations of the system

Next we consider the basic equations of the system described by the transfer matrix equations (1.48):

$$\widehat{\mathbf{IU}} \cdot \widehat{\mathbf{Z}} = \overset{\circ}{\mathbf{Z}} \quad (2.2)$$

where

$$\widehat{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_L \\ \mathbf{Z}_A \end{bmatrix} \quad (2.3)$$

\mathbf{Z}_L , \mathbf{Z}_A are the vectors of displacements and forces at the end and at the beginning of the element, respectively, and $\overset{\circ}{\mathbf{Z}}$ is the loading vector of the element from Eq. (1.46) at $x = l$.

To insert the system of basic equations (2.2) into the system of Eq. (2.1), the GNU Octave functions `ysplvmhvI.m` (p. 244), `yzhqz.m` (p. 242), `yzfzv.m` (p. 243), `spInsertBtoA.m` (p. 241), `InsertBtoA.m` (p. 242) are used. The procedure of inserting is given in excerpt 2.1 from the program.

¹<http://www.gnu.org/software/octave/doc/interpreter/>. Web. 08 August 2013.

²<http://www.network-theory.co.uk/docs/octave3/index.html>. Web. 08 August 2013.

³<http://users.powernet.co.uk/kienzle/octave/matcompat/>. Web. 08 August 2013.

Program excerpt 2.1 (The basic equations. `LaheFrameDFIm.m`)

```

IIv=0;
IJv=0;
%
for i=1:NEARV
    krda=i;
    EI=selem(i,13); % from topology
    EA=selem(i,14); % " "
    GAR=selem(i,15); % " "
    Li=lvarras(i,1);
    qx=qxZ(i,1);
    qz=qzZ(i,1);
    aLx=aLXx(i,1);
    Fz=FZz(i,1);
    Fx=FZx(i,1);
%----- Transfer matrix equation -----
    spvF=yvspvmhvi(baasi0, Li, Li, EA, GAR, EI);
    vB=yzhqz(baasi0, Li, qx, qz, EA, EI);
    vFz=yzfzv(baasi0, Li, aLx, Fx, Fz, EA, EI);
    vB=vB+vFz;
    IIv=krda*6-5;
    IJv=krda*12-11;
    spA=spInsertBtoA(spA, IIv, IJv, spvF);
    B=InsertBtoA(B, NNK, 1, IIv, 1, vB, 6, 1);
endfor
% here NEARV --- number of elements,
% baasi0 --- scaling multiplier for displacements.

```

2.2 Compatibility equations of displacements

Consider a system of compatibility equation sets of the joint displacements shown in excerpts 2.1 and 2.2 from the computing diary.

Computing diary excerpt 2.1 (`yvspESTframe2LaheWFI.m`)

```

=====
Element | Displacements | Joint | Axial, shear, moment hinges
No      | u_   w_   fi_ | (node) | 0 - hinge 'false'
        | indexes | No     | 1 - hinge 'true'
-----
    1    1    2    3    2        0    0    0
    2    19   20   21   2        0    0    0

```

In excerpt 2.1 from the computing diary, two elements, 1 and 2, at joint node 2 are shown. The displacements of element 1 at node 2 are Z_1, Z_2, Z_3 and the displacements of element 2 at node 2 are Z_{19}, Z_{20}, Z_{21} . The joint at node 2 is rigid (see Fig. 1.12).

We now connect elements 1 and 2, the end displacements of which are

$$\mathbf{v}_L^{(1)} = \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix}, \quad \mathbf{v}_A^{(2)} = \begin{bmatrix} u_A^{(2)} \\ w_A^{(2)} \\ \varphi_A^{(j)} \end{bmatrix} \quad (2.4)$$

The compatibility equations at node 2 in global coordinates:

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{v}_L^{(1)} - \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{v}_A^{(2)} = 0 \quad (2.5)$$

where the transformation matrices are expressed as

$$\mathbf{T}_{3 \times 3}^{(1)} = \begin{bmatrix} \cos \alpha_1 & -\cos \beta_1 & 0 \\ \cos \beta_1 & \cos \alpha_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_{3 \times 3}^{(2)} = \begin{bmatrix} \cos \alpha_2 & -\cos \beta_2 & 0 \\ \cos \beta_2 & \cos \alpha_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Computing diary excerpt 2.2 (`yspESTframe2LaheWFI.m`)

```

=====
Element | Displacements | Joint | Axial, shear, moment hinges
No      | u_   w_   fi_ | (node) | 0 - hinge 'false'
        | indexes | No     | 1 - hinge 'true'
-----
      2   13   14   15     4         0     0     0
      3   25   26   27     4         0     0     0
      4   43   44   45     4         0     0     1

```

In excerpt 2.2 from the computing diary, three elements – 2, 3 and 4 – at joint node 4 are shown. The displacements of element 2 at node 4 are Z_{13} , Z_{14} , Z_{15} , and the displacements of element 4 at node 4 are Z_{43} , Z_{44} , Z_{45} . Element 4 has a *bending moment hinge* at node 4 (see Fig. 1.12).

Now we connect elements 2 and 4, the end displacements of which are

$$\mathbf{v}_{\mathbf{L}}^{(2)} = \begin{bmatrix} u_L^{(2)} \\ w_L^{(2)} \end{bmatrix} \quad \text{and} \quad \mathbf{v}_{\mathbf{A}}^{(4)} = \begin{bmatrix} u_A^{(4)} \\ w_A^{(4)} \end{bmatrix} \quad (2.7)$$

The compatibility equations for the pair displacements of elements 2 and 4:

$$\mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{v}_{\mathbf{L}}^{(2)} - \mathbf{T}_{2 \times 2}^{(4)} \cdot \mathbf{v}_{\mathbf{A}}^{(4)} = 0 \quad (2.8)$$

where the transformation matrices (see Eq. (A.25)) have the form

$$\mathbf{T}_{2 \times 2}^{(2)} = \begin{bmatrix} \cos \alpha_2 & -\cos \beta_2 \\ \cos \beta_2 & \cos \alpha_2 \end{bmatrix}, \quad \mathbf{T}_{2 \times 2}^{(4)} = \begin{bmatrix} \cos \alpha_4 & -\cos \beta_4 \\ \cos \beta_4 & \cos \alpha_4 \end{bmatrix} \quad (2.9)$$

There may be a number of displacement pairs of elements: 2–3, 2–4, 3–4, The set of pairs 2–3, 2–4, 3–4 is linearly independent. We need a maximal linearly independent subset of the set pairs. A linearly independent set is called maximal if each of its proper superset is linearly dependent [OUD10], [KB95]. Compatibility requirements should be satisfied in the compatibility method, called also the matrix force method. We should note that at rigid joint frames, the multiple hinge is equivalent to $n-1$ simple hinges (n is the number of members connected in the joint). Here, compatibility equations must be verified to meet all the criteria of transitivity⁴. We consider pairs 2–3, 2–4 or pairs 2–3, 3–4, which are linearly independent sets.

⁴<http://en.wikipedia.org/wiki/Transitivity>.

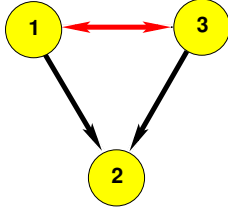


Figure 2.1. Transitivity

A relation between element end displacements is *transitive* whenever a displacement at bar 1 end point is equal to a displacement at bar 2 starting point, and a displacement at bar 3 end point is equal to a displacement at bar 2 starting point. Then the displacement at bar 1 end point is equal to the displacement at bar 3 end point (see Fig. 2.1).

Let us now consider the compatibility of displacements and rotations at the joints.

Displacements at joint A3 (see Fig. 2.2). Here, two elements are connected at a rigid joint.

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(2)} \\ w_A^{(2)} \\ \varphi_A^{(2)} \end{bmatrix} = 0 \quad (2.10)$$

or

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{v}_L^{(1)} - \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{v}_A^{(2)} = 0 \quad (2.11)$$

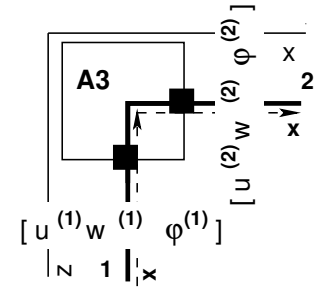


Figure 2.2. Compatibility at rigid joint A3

Displacements at joint A2 (see Fig. 2.3). Here, two elements are connected at a hinged joint.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(2)} \\ w_A^{(2)} \end{bmatrix} = 0 \quad (2.12)$$

or

$$\mathbf{T}_{2 \times 2}^{(1)} \cdot \mathbf{v}_{2L}^{(1)} - \mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{v}_{2A}^{(2)} = 0 \quad (2.13)$$

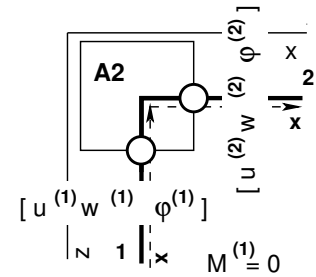


Figure 2.3. Compatibility at hinged joint A2

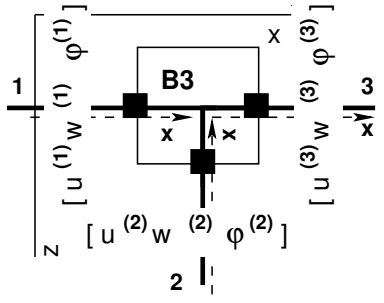
Displacements at joint B3 (see Fig. 2.4). Three elements are connected at a rigid joint. The displacement compatibility for elements 1–2 is given as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(2)} \\ w_A^{(2)} \\ \varphi_A^{(2)} \end{bmatrix} = 0 \quad (2.14)$$

or

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{v}_L^{(1)} - \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{v}_A^{(2)} = 0 \quad (2.15)$$

The displacement compatibility for elements 2–3 is given as



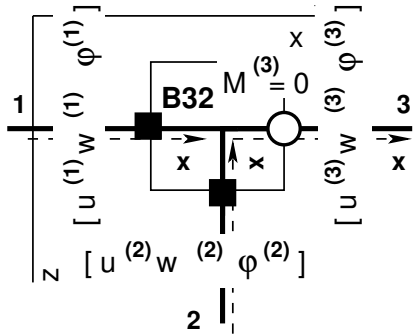
$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(2)} \\ w_L^{(2)} \\ \varphi_L^{(2)} \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(3)} \\ w_A^{(3)} \\ \varphi_A^{(3)} \end{bmatrix} = 0 \quad (2.16)$$

Figure 2.4. Compatibility at rigid joint B3

or

$$\mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{v}_L^{(2)} - \mathbf{T}_{3 \times 3}^{(3)} \cdot \mathbf{v}_A^{(3)} = 0 \quad (2.17)$$

Displacements at joint B32 (see Fig. 2.5). Three elements are gathered together at a joint with rigid and pin connection. Elements 2–3 are connected at the joint by a hinge.



$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u_L^{(2)} \\ w_L^{(2)} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(3)} \\ w_A^{(3)} \end{bmatrix} = 0 \quad (2.18)$$

or

$$\mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{v}_{2L}^{(2)} - \mathbf{T}_{2 \times 2}^{(3)} \cdot \mathbf{v}_{2A}^{(3)} = 0 \quad (2.19)$$

Figure 2.5. Compatibility at rigid & pin joint B32

Elements 1–2 are connected rigidly at the joint. Their displacement compatibility equations are

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(2)} \\ w_L^{(2)} \\ \varphi_L^{(2)} \end{bmatrix} = 0 \quad (2.20)$$

or

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{v}_L^{(1)} - \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{v}_L^{(2)} = 0 \quad (2.21)$$

Displacements at joint B2 (see Fig. 2.6). Three elements – 1, 2, and 3 – are connected at a pin joint. Elements 1–2 are connected by a hinge.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_L^{(1)} \\ w_L^{(1)} \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u_A^{(2)} \\ w_A^{(2)} \end{bmatrix} = 0 \quad (2.22)$$

or

$$\mathbf{T}_{2 \times 2}^{(1)} \cdot \mathbf{v}_{2L}^{(1)} - \mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{v}_{2A}^{(2)} = 0 \quad (2.23)$$

Elements 2–3 are also connected by a hinge.

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} u_L^{(2)} \\ w_L^{(2)} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_A^{(3)} \\ w_A^{(3)} \end{bmatrix} = 0 \quad (2.24)$$

or

$$\mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{v}_{2L}^{(2)} - \mathbf{T}_{2 \times 2}^{(3)} \cdot \mathbf{v}_{2A}^{(3)} = 0 \quad (2.25)$$

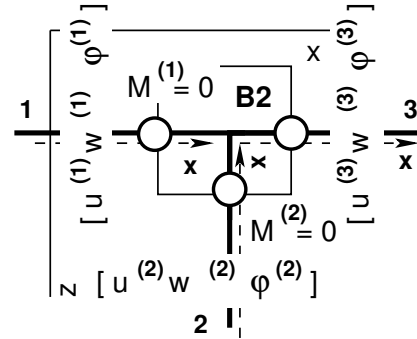


Figure 2.6. Compatibility at pin joint B2

The transformation matrices $\mathbf{T}_{3 \times 3}^{(i)}$ and $\mathbf{T}_{2 \times 2}^{(i)}$ (in the program as *SpTM3x3* and *SpTM2x2*) are represented with the functions [SpTeisendusMaatriks.m](#) and [SpTeisendusMaatriks2x2.m](#), respectively. The program selects the row index and finds column and element indexes (see excerpts 2.1 and 2.2 from the computing diary). Taking into account the expressions $SpTM3x3m = -SpTM3x3$ and $SpTM2x2m = -SpTM2x2$, the compatibility equations of the displacements at a node can be inserted into Eq. (2.1). The program inserts the transformation matrices with the commands `cmd = spA=spInsertBtoA(spA,34,43,SpTM3x3)`; and `cmd = spA=spInsertBtoA(spA,34,25,SpTM3x3m)`; (see excerpt 2.3).

Computing diary excerpt 2.3 (`spESTframe93LaheWFI.m`)

```

=====
---- Sparse matrix instantiation -----
    spA=sparse(NNK,NNK)

spA =
Compressed Column Sparse (rows = 67, cols = 67, nnz = 0 [0%])

---- Right-hand side of the equations (RHS). -----
    B=zeros(NNK,1);

#=====
---- Writing basic equations of a frame ----
#=====

---- Basic equations are inserted into spA -----
rows = rows_of_basic_equations: 30
col = cols_of_basic_equations: 60
spA_nnz = non_zero_elements_in_basic_equations: 95

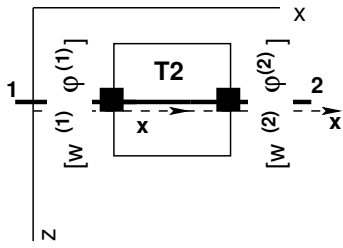
#=====
Compatibility equations of displacements at nodes
From_row = Compatibility equations begin from row: 31
#=====
Node = 1
Node = 2
cmd = spA=spInsertBtoA(spA,31,19,SpTM3x3);
cmd = spA=spInsertBtoA(spA,31,1,SpTM3x3m);
Node = 3
Node = 4
cmd = spA=spInsertBtoA(spA,34,43,SpTM3x3);
cmd = spA=spInsertBtoA(spA,34,25,SpTM3x3m);
cmd = spA=spInsertBtoA(spA,37,43,SpTM3x3);
cmd = spA=spInsertBtoA(spA,37,13,SpTM3x3m);
Node = 5
Node = 6
cmd = spA=spInsertBtoA(spA,40,55,SpTM3x3);
cmd = spA=spInsertBtoA(spA,40,37,SpTM3x3m);
----
spA_rows = 42
spA_cols = 60
spA_nnz = non_zero_elements_in_spA: 129
--- Compatibility equations of displacements are inserted into spA ---
compatibility_equations_rows = 12
non_zero_elements_in_compatibility_equations = 34
=====

```

2.2.1 Compatibility conditions at beam joints

The beam considered is exposed to a shear force and bending moment, and accordingly has the deflection w and slope φ .

The two beam elements in Fig. 2.7 have a rigid connection:



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_A^{(2)} \\ \varphi_A^{(2)} \end{bmatrix} = 0 \quad (2.26)$$

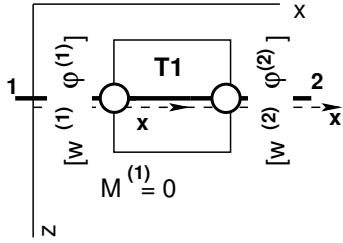
or

$$\mathbf{I}_{2 \times 2} \cdot \mathbf{v}_L^{(1)} - \mathbf{I}_{2 \times 2} \cdot \mathbf{v}_A^{(2)} = 0 \quad (2.27)$$

where $\mathbf{I}_{2 \times 2}$ is a unit matrix.

Figure 2.7. Compatibility at rigid joint T2

In Fig. 2.8, the two beam elements have a hinged connection:



$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} w_L^{(1)} \\ \varphi_L^{(1)} \end{bmatrix} - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} w_A^{(2)} \\ \varphi_A^{(2)} \end{bmatrix} = 0 \quad (2.28)$$

or

$$\mathbf{T}_1^o \cdot \mathbf{v}_L^{(1)} - \mathbf{T}_2^o \cdot \mathbf{v}_A^{(2)} = 0 \quad (2.29)$$

Figure 2.8. Compatibility at pin joint T1

2.3 Equations of joint equilibrium

With joint equilibrium equations, the same transformation matrices of Eqs. (2.6) and (2.9) are used as by writing the compatibility equations. The sum of the internal reactions acting on either side of the cut surface (section) is equal to zero:

$$\begin{bmatrix} N_{element}^{(i)} \\ Q_{element}^{(i)} \\ M_{element}^{(i)} \end{bmatrix} + \begin{bmatrix} N_{node}^{(i)} \\ Q_{node}^{(i)} \\ M_{node}^{(i)} \end{bmatrix} = 0, \quad \text{that is,} \quad \begin{bmatrix} N_{element}^{(i)} \\ Q_{element}^{(i)} \\ M_{element}^{(i)} \end{bmatrix} = - \begin{bmatrix} N_{node}^{(i)} \\ Q_{node}^{(i)} \\ M_{node}^{(i)} \end{bmatrix} \quad (2.30)$$

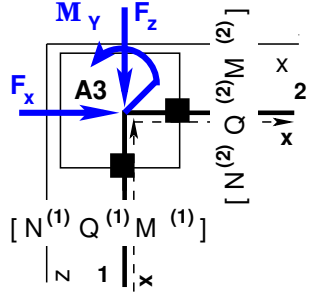
The internal reactions acting on either side of the cut surface are equal and opposite. We use Sign Convention 2 for the frame elements.

The equilibrium equations set the sum of the internal reactions in Fig. 1.12 (contact forces⁵) equal to the externally applied loads \mathcal{F}_{ext} at joints or node points. We will call the internal reactions *contact forces and contact moments*.

Equilibrium of joint A3 (see Fig. 2.9). The sum of the contact forces $N_L^{(1)}$, $Q_L^{(1)}$,

⁵A *contact force* is a force that acts at the points of contact between two objects [Rand07].

$M_L^{(1)}$ of element 1 and $N_L^{(2)}, Q_L^{(2)}, M_L^{(2)}$ of element 2 is equal to the external forces F_x, F_z, \mathcal{M}_y at rigid joint A3, respectively.



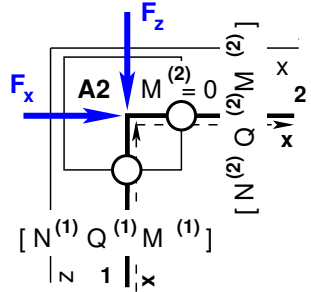
$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \\ M_L^{(1)} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(2)} \\ Q_A^{(2)} \\ M_A^{(2)} \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.31)$$

Figure 2.9. Equilibrium of joint A3

or

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{s}_L^{(1)} + \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{s}_A^{(2)} = \mathcal{F}_{A3} \quad (2.32)$$

Equilibrium of joint A2 (see Fig. 2.10). The sum of the contact forces $N_L^{(1)}, Q_L^{(1)}$ of element 1 and $N_L^{(2)}, Q_L^{(2)}$ of element 2 is equal to the external forces F_x, F_z at hinged joint A3, respectively.



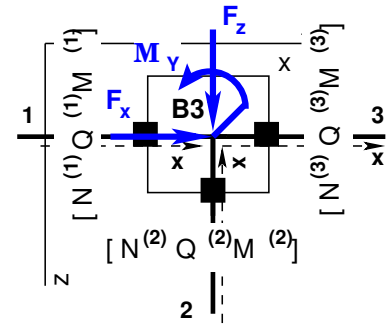
$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(2)} \\ Q_A^{(2)} \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (2.33)$$

Figure 2.10. Equilibrium of joint A2

or

$$\mathbf{T}_{2 \times 2}^{(1)} \cdot \mathbf{s}_{2L}^{(1)} + \mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{s}_{2A}^{(2)} = \mathcal{F}_{A2} \quad (2.34)$$

Equilibrium of joint B3 (see Fig. 2.11). The sum of the contact forces $N_L^{(1)}, Q_L^{(1)}, M_L^{(1)}$ of element 1, $N_L^{(2)}, Q_L^{(2)}, M_L^{(2)}$ of element 2, and $N_A^{(3)}, Q_A^{(3)}, M_A^{(3)}$ of element 3 is equal to the external forces F_x, F_z, \mathcal{M}_y at rigid joint B3, respectively.



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \\ M_L^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(2)} \\ Q_L^{(2)} \\ M_L^{(2)} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(3)} \\ Q_A^{(3)} \\ M_A^{(3)} \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.35)$$

Figure 2.11. Equilibrium of joint B3

or

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{s}_L^{(1)} + \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{s}_L^{(2)} + \mathbf{T}_{3 \times 3}^{(3)} \cdot \mathbf{s}_A^{(3)} = \mathcal{F}_{B3} \quad (2.36)$$

Equilibrium of joint B2 (see Fig. 2.12). The sum of the contact forces $N_L^{(1)}$, $Q_L^{(1)}$ of element 1, $N_L^{(2)}$, $Q_L^{(2)}$ of element 2, and $N_A^{(3)}$, $Q_A^{(3)}$ of element 3 is equal to the external forces F_x , F_z at hinged joint B2, respectively.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} N_L^{(2)} \\ Q_L^{(2)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(3)} \\ Q_A^{(3)} \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (2.37)$$

or

$$\begin{aligned} \mathbf{T}_{2 \times 2}^{(1)} \cdot \mathbf{s}_{2L}^{(1)} + \mathbf{T}_{2 \times 2}^{(2)} \cdot \mathbf{s}_{2L}^{(2)} \\ + \mathbf{T}_{2 \times 2}^{(3)} \cdot \mathbf{s}_{2A}^{(3)} = \mathcal{F}_{B2} \end{aligned} \quad (2.38)$$

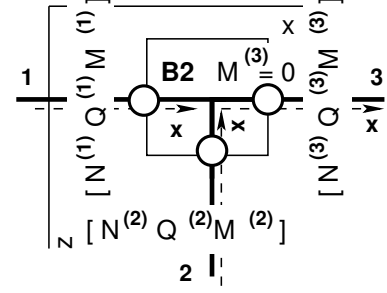


Figure 2.12. Equilibrium of joint B2

Equilibrium of joint B32 (see Fig. 2.13). The sum of the contact forces $N_L^{(1)}$, $Q_L^{(1)}$, $M_L^{(1)}$ of element 1, $N_L^{(2)}$, $Q_L^{(2)}$, $M_L^{(2)}$ of element 2, and $N_A^{(3)}$, $Q_A^{(3)}$ of element 3 is equal to the external forces F_x , F_z , \mathcal{M}_y at rigid-hinged joint B32, respectively.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \\ M_L^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_L^{(2)} \\ Q_L^{(2)} \\ M_L^{(2)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} N_A^{(3)} \\ Q_A^{(3)} \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.39)$$

or

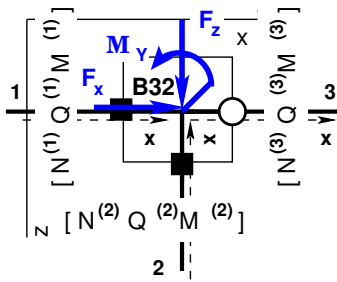
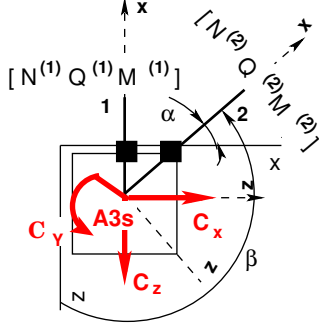


Figure 2.13. Equilibrium of joint B32

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{s}_L^{(1)} + \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{s}_L^{(2)} + \mathbf{T}_{3 \times 2}^{(3)} \cdot \mathbf{s}_{2A}^{(3)} = \mathcal{F}_{B32} \quad (2.40)$$

In Fig. 2.13, the three elements are gathered together at the joint with a rigid and pin connection. Unlike compatibility equations, the number of equilibrium equations at the joint does not depend on the number of elements at the joint (see p. 38).

Equilibrium of joint A3s (see Fig. 2.14). The external reactions C_x, C_z, C_y at the support node are described in global coordinates. The direction of external reactions at the joint is opposite to the direction of internal reactions at the joint and equal to the direction of the external forces F_x, F_z, \mathcal{M}_y (not shown in Fig. 2.14) at the joint.

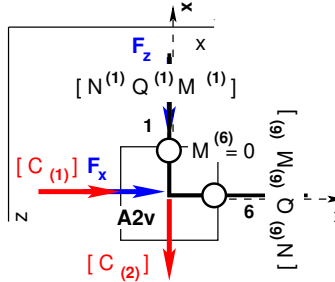


$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(1)} \\ Q_A^{(1)} \\ M_A^{(1)} \end{bmatrix} + \begin{bmatrix} \cos \alpha & -\cos \beta & 0 \\ \cos \beta & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(2)} \\ Q_A^{(2)} \\ M_A^{(2)} \end{bmatrix} - \begin{bmatrix} C_x \\ C_z \\ C_y \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.41)$$

or

Figure 2.14. Equilibrium of support node A3s

$$\mathbf{T}_{3 \times 3}^{(1)} \cdot \mathbf{s}_A^{(1)} + \mathbf{T}_{3 \times 3}^{(2)} \cdot \mathbf{s}_A^{(2)} - \mathbf{I}_{3 \times 3} \cdot \mathcal{C}_{A3s} = \mathcal{F}_{A3s} \quad (2.42)$$

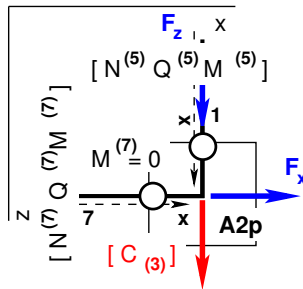


$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} N_L^{(1)} \\ Q_L^{(1)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(6)} \\ Q_A^{(6)} \end{bmatrix} - \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (2.43)$$

or

Figure 2.15. Equilibrium of support node A2v

$$\mathbf{T}_{2 \times 2}^{(1)} \cdot \mathbf{s}_{2L}^{(1)} + \mathbf{T}_{2 \times 2}^{(6)} \cdot \mathbf{s}_{2A}^{(6)} - \mathcal{C}_{A2v} = \mathcal{F}_{A2v} \quad (2.44)$$



$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} N_L^{(5)} \\ Q_L^{(5)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} N_A^{(7)} \\ Q_A^{(7)} \end{bmatrix} - \begin{bmatrix} \dots \\ C_3 \end{bmatrix} = \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (2.45)$$

or

Figure 2.16. Equilibrium of support node A2p

$$\mathbf{T}_{2 \times 2}^{(5)} \cdot \mathbf{s}_{2L}^{(5)} + \mathbf{T}_{2 \times 2}^{(7)} \cdot \mathbf{s}_{2A}^{(7)} - \mathcal{C}_{A2p} = \mathcal{F}_{A2p} \quad (2.46)$$

The program selects the row index and finds column and element indexes (see excerpts 2.1 and 2.2 from the computing diary). It inserts the joint equilibrium equations into Eq. (2.1). The transformation matrix is inserted with the command `cmd = spA=spInsertBtoA(spA,45,22,SpTM3x3)`; (see excerpt 2.4).

Computing diary excerpt 2.4 (`spESTframe93LaheWFI.m`)

```

=====
Compatibility equations of displacements are inserted into spA
compatibility_equations_rows = 12
non_zero_elements_in_compatibility_equations = 34
=====
Joint equilibrium equations at nodes
From_rows = Joint equilibrium equations begin from row: 43
=====
Node = 1
Number of reactions at the node (node_no, reactions): 1, 2
cmd = spA=spInsertBtoA(spA,43,10,SpTM2x2);
cmd = spA=spInsertBtoA(spA,43,61,SpTM2x2xz);
-----

Node = 2
cmd = spA=spInsertBtoA(spA,45,22,SpTM3x3);
cmd = spA=spInsertBtoA(spA,45,4,SpTM3x3);
Nodal forces at the node
cmd = B(45:47,1)=sSolmF(1:3,2);
-----

Node = 3
Number of reactions at the node (node_no, reactions): 3, 3
cmd = spA=spInsertBtoA(spA,48,34,SpTM3x3);
cmd = spA=spInsertBtoA(spA,48,63,SpTM3x3xz);
-----

Node = 4
cmd = spA=spInsertBtoA(spA,51,46,SpTM3x3);
cmd = spA=spInsertBtoA(spA,51,28,SpTM3x3);
cmd = spA=spInsertBtoA(spA,51,16,SpTM3x3);
Nodal forces at the node
cmd = B(51:53,1)=sSolmF(1:3,4);
-----

Node = 5
Number of reactions at the node (node_no, reactions): 5, 2
cmd = spA=spInsertBtoA(spA,54,52,SpTM2x2);
cmd = spA=spInsertBtoA(spA,54,66,SpTM2x2xz);
-----

Node = 6
cmd = spA=spInsertBtoA(spA,56,58,SpTM3x3);
cmd = spA=spInsertBtoA(spA,56,40,SpTM3x3);
Nodal forces at the node
cmd = B(56:58,1)=sSolmF(1:3,6);
=====
spA_rows = 58
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 172

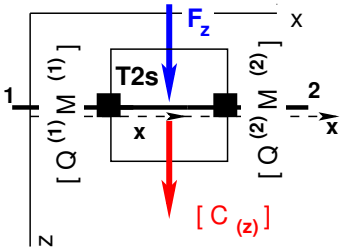
----Equilibrium equations are inserted into spA ----

equilibrium_equations_rows = 16
non_zero_elements_in_equilibrium_equations = 43
=====

```

2.3.1 Equilibrium equations of beam joints

Equilibrium equations of joint T2s. The equilibrium equations of the beam elements at joint or node points set the internal and external reactions \mathcal{C}_{T2s} equal to the sum of the applied loads (see Fig. 2.17).



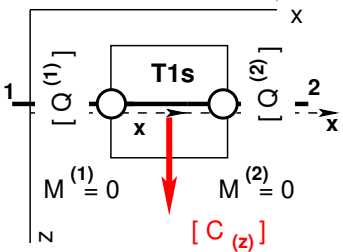
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_L^{(1)} \\ M_L^{(1)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_A^{(2)} \\ M_A^{(2)} \end{bmatrix} - \begin{bmatrix} C_z \\ C_y \end{bmatrix} = \begin{bmatrix} F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.47)$$

or

$$\mathbf{T}_1 \cdot \mathbf{s}_{2L}^{(1)} + \mathbf{T}_2 \cdot \mathbf{s}_{2A}^{(2)} - \mathcal{C}_{T2s} = \mathcal{F}_{T2s} \quad (2.48)$$

Figure 2.17. Equilibrium of joint T2s

Equilibrium equations of joint T1s. The external force F_z at the joint is not shown in Fig. 2.18.



$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_L^{(1)} \\ M_L^{(1)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_A^{(2)} \\ M_A^{(2)} \end{bmatrix} - \begin{bmatrix} C_z \\ C_y \end{bmatrix} = \begin{bmatrix} F_z \\ \mathcal{M}_y \end{bmatrix} \quad (2.49)$$

or

$$\mathbf{T}_1^o \cdot \mathbf{s}_{2L}^{(1)} + \mathbf{T}_2^o \cdot \mathbf{s}_{2A}^{(2)} - \mathcal{C}_{T1s} = \mathcal{F}_{T1s} \quad (2.50)$$

Figure 2.18. Equilibrium of joint T1s

2.4 Side conditions

The program selects the row index and finds column and element indexes (see excerpt 2.5). It inserts the side conditions of Eq. (2.51) into Eq. (2.1). The equations are inserted with the commands $cmd = spA = spSisestaArv(spA, 59, 12, 1)$; and $cmd = spA = spSisestaArv(spA, 60, 54, 1)$; (see excerpt 2.6).

Computing diary excerpt 2.5 (`spESTframe93LaheWFI.m`)

```
=====
Element | Internal      | Joint | Axial, shear, moment hinges
  No    | reactions     | (node) | 0 - hinge 'false'
        | N_   Q_   M_ | No     | 1 - hinge 'true'
        | indexes      |       |
-----
```

Element No	Internal reactions indexes	Joint (node) No	Axial	shear	moment hinges
1	10 11 12	1	0	0	1
1	4 5 6	2	0	0	0
2	22 23 24	2	0	0	0
3	34 35 36	3	0	0	0
2	16 17 18	4	0	0	0
3	28 29 30	4	0	0	0
4	46 47 48	4	0	0	0
5	52 53 54	5	0	0	1
4	40 41 42	6	0	0	0
5	58 59 60	6	0	0	0

$$\mathbf{M}_{12}^{(1)} = 0 \quad \text{and} \quad \mathbf{M}_{54}^{(5)} = 0 \quad (2.51)$$

The bending moment at the hinge location is zero exclusive of a plastic hinge⁶.

Computing diary excerpt 2.6 (`spESTframe93LaheWFI.m`)

```
=====
-----Equilibrium equations are inserted into spA

equilibrium_equations_rows = 16
non_zero_elements_in_equilibrium_equations = 43

#=====
Side conditions
From_rows = Side conditions begin from row: 59
#=====
cmd = spA=spSisestaArv(spA,59,12,1);
cmd = spA=spSisestaArv(spA,60,54,1);
-----

spA_rows = 60
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 174

-----Side conditions are inserted into spA

side_condition_rows = 2
non_zero_elements_in_side_condition = 2

#=====
```

⁶At the formation of a plastic hinge the moment at the ends of an element remains constant at the value M_{pl} (see Fig. 7.5 and section 8.1).

2.5 Restrictions on support displacements

Restrictions on the support displacements (see Fig. 1.10) of the system (2.1) are given in excerpt 2.7 from the computing diary and are described in global coordinates.

Computing diary excerpt 2.7 (`spESTframe93LaheWFI.m`)

```
=====
Restrictions on support displacements (on - 1, off - 0)
  No   Node No   u   w   fi
-----
  1     1     1   1   0
  2     3     1   1   1
  3     5     1   1   0
-----
```

The global displacements of the structure at a support node are obtained by transforming the element displacements to the global coordinate system.

The program selects the row index and finds column and element indexes (see excerpts 2.7 and 2.8). Restrictions on the support displacements are inserted into Eq. (2.1).

Computing diary excerpt 2.8 (`spESTframe93LaheWFI.m`)

```
=====
Support | Restrictions | Element | Displacements
node   |(on - 1, off - 0) | No     | u_  w_  fi_
No     | u_  w_  fi_   |       | indexes
-----
  1     1   1   0     1     7   8   9
  3     1   1   1     3    31  32  33
  5     1   1   0     5    49  50  51
-----
```

The program inserts the displacement component of an element at a support into Eq. (2.1) with the command $cmd = spA = spInsertBtoA(spA, 61, 7, SpTMUv)$; (see excerpt 2.9) where $SpTUv = SpToeSiirdeUvektor(NSARV, NEARV, VarrasS, krdn, selem)$, $SpTWv = SpToeSiirdeWvektor(NSARV, NEARV, VarrasS, krdn, selem)$, $SpTFiV = SpToeSiirdeFiVektor(1)$.

The support shift is inserted with the command $cmd = B(61, 1) = tSiire(1, 1, 1)$; The program multiplies the support shifts by the basic stiffness ⁷, $i_0 = EI_{basic}/l_{basic}$.

Computing diary excerpt 2.9 (`spESTframe93LaheWFI.m`)

```
#####
-----Side conditions are inserted into spA -----

side_condition_rows = 2
non_zero_elements_in_side_condition = 2
```

⁷Basic stiffness – the scaling multiplier for displacements.

```

=====
Restrictions on support displacements
Restrictions on support displacements begin from row: 61
=====

cmd = spA=spInsertBtoA(spA,61,7,SpTUv);
Support shift tSiire(1,1,1) = 0 in x direction at node 1
cmd = B(61,1)=tSiire(1,1,1);

cmd = spA=spInsertBtoA(spA,62,7,SpTWv);
Support shift tSiire(2,1,1) = 0 in z direction at node 1
cmd = B(62,1)=tSiire(2,1,1);

cmd = spA=spInsertBtoA(spA,63,31,SpTUv);
Support shift tSiire(1,1,3) = 0 in x direction at node 3
cmd = B(63,1)=tSiire(1,1,3);

cmd = spA=spInsertBtoA(spA,64,31,SpTWv);
Support shift tSiire(2,1,3) = 0 in z direction at node 3
cmd = B(64,1)=tSiire(2,1,3);

cmd = spA=spInsertBtoA(spA,65,31,SpTFiV);
Support shift tSiire(3,1,3) = 0 in y direction at node 3
cmd = B(65,1)=tSiire(3,1,3);

cmd = spA=spInsertBtoA(spA,66,49,SpTUv);
Support shift tSiire(1,1,5) = 0 in x direction at node 5
cmd = B(66,1)=tSiire(1,1,5);

cmd = spA=spInsertBtoA(spA,67,49,SpTWv);
Support shift tSiire(2,1,5) = 0 in z direction at node 5
cmd = B(67,1)=tSiire(2,1,5); \Eq.~
-----

spA_rows = 67
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 181

-----Restrictions equations on displacements are inserted into spA

restrictions_equations_rows = 7
non_zero_elements_in_restrictions_equations = 7

=====
spA_rank = 67
spA =

Compressed Column Sparse (rows = 67, cols = 67, nnz = 181 [4%])
=====

```

2.6 Initial parameter vector for an element

Consider again the assembled system of equations (2.1) of the boundary value problem. It is necessary but not sufficient for a well posed boundary problem that the rank of equations (2.1) be equal to the number of linear equations: $N_{eqs} = rank(spA)$ (see excerpt 2.10).

Computing diary excerpt 2.10 (`spESTframe93LaheWFI.m`)

```
=====
spA_rank = 67
spA =
Compressed Column Sparse (rows = 67, cols = 67, nnz = 181 [4%])
=====
```

Solving the sparse system of equations. To solve the sparse system of equations (2.1), the command

$$\mathbf{Z} = \mathbf{spA} \backslash \mathbf{B} \quad (2.52)$$

is made use of (see excerpt 2.11).

Computing diary excerpt 2.11 (`spESTframe93LaheWFI.m`)

```
=====
spA =
Compressed Column Sparse (rows = 67, cols = 67, nnz = 181 [4%])
=====
Solving the sparse system of equations
Z=spA\B;
Support_reactions = Support reactions begin from row: 61
=====
No      Z
1      -9.857e-11      7      -0.000e+00      13      -1.686e+02      19      -1.686e+02
2      -1.706e+02      8      0.000e+00      14      -2.623e+01      20      -2.623e+01
3      6.483e+00      9      4.246e+01      15      8.975e+00      21      6.483e+00
4      -2.267e+01      10     2.267e+01      16     -1.372e+01      22      1.618e+01
5      -1.285e+01      11     1.285e+01      17     4.616e+00      23     -2.043e+01
6      -7.196e+01      12     0.000e+00      18     4.209e+01      24     7.196e+01
...     ...
61     1.285e+01      63     2.728e+01      66     4.948e+01
62     -2.267e+01      64     4.870e+00      67     1.800e+00
65     -1.026e+02
=====
```

Support reactions are in global coordinates.

It is necessary to verify the structure for a static equilibrium:

- the sum of the external forces (load forces and support reactions) is zero,
- the sum of the moments of all the external forces about any point is zero.

Dividing now each of the displacements and rotations by the scaling multiplier, $i_0 = EI_{basic}/l_{basic}$, we obtain an unscaled initial parameter vector (Sign Convention 2) in local coordinates (see excerpt 2.12).

Computing diary excerpt 2.12 (`spESTframe93LaheWFI.m`)

```
=====
Unscaled initial parameter vector
Element   u           w           fi           N           Q           M
-----
1   -0.000e+00   0.000e+00   1.189e-02   22.670   12.850   0.000
2   -4.721e-02  -7.343e-03   1.815e-03   16.182  -20.426   71.959
3   -0.000e+00   0.000e+00   0.000e+00   -4.870   27.275  -102.642
4   -4.721e-02   7.343e-03   2.513e-03   39.371   -7.946   46.197
5   -2.192e-15   4.778e-02   2.340e-04   -1.800  -40.125  -26.181
=====
```

2.7 Element displacements and forces

To determine element displacements and forces, the transfer matrix is used. Calculating the internal reaction^{8 9 10} of Eq. (2.53) for drawing internal forces diagrams, the calculating trick is done (see Fig. 2.19).

$$\mathbf{Z}_L(\mathbf{x}) = \mathbf{U} \cdot \mathbf{Z}_A + \overset{\circ}{\mathbf{Z}} \quad (2.53)$$

where

\mathbf{Z}_L is the vector of displacements and forces at \mathbf{x} ,

\mathbf{Z}_A – unscaled initial parameter vector (Sign Convention 2) in local coordinates (see excerpt 2.12),

\mathbf{U} – transfer matrix in Eq. (1.45),

$\overset{\circ}{\mathbf{Z}}$ – loading vector in Eq. (1.46),

The scaling multiplier for element displacements $i_0 = 1$ ($i_0 = EI/l$).

Calculations of the internal reactions (internal forces correspond to Sign Convention 1) are presented in excerpt 2.13.

⁸WP 960 Technical Description: ... Two force gauges determine *the internal reactions*... [WP960].

⁹http://engr.bd.psu.edu/rxm61/213/Beams_overview.ppt. Web. 20 August 2013.

¹⁰Shear and Moment Diagrams for Frames: ... cut the frame into its component members and find *the internal reactions* [MPHS].

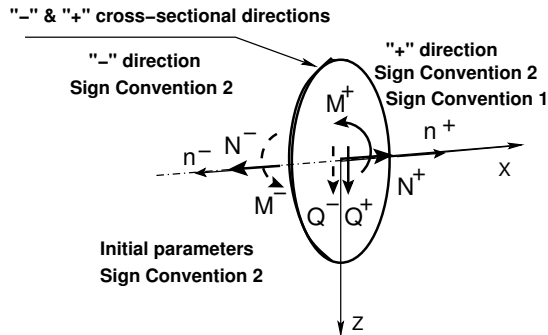


Figure 2.19. Cross-sectional directions

In Fig. 2.19, minus and plus cross-sectional directions are shown. The sign of the internal forces corresponds to Sign Convention 1 of the internal reactions. To calculate an internal reaction for drawing internal forces diagrams, a calculating trick is done. As the sign of the initial parameters corresponds to Sign Convention 2, the internal reactions are calculated at $x = 0$ cross-section (Eq. (2.53)).

Computing diary excerpt 2.13 (`spESTframe93LaheWFI.m`)

```
#----- Element displacements and forces -----
for i=1:NEARV
    krda=i;
    vF=zeros(6,12);
    EI=selem(i,13); % from topology
    EA=selem(i,14); % " "
    GAR=selem(i,15); % " "
    Li=lvarras(i,1);
    qx=qxZ(i,1);
    qz=qzZ(i,1);
    aLx=aLXx(i,1);
    Fz=FZz(i,1);
    Fx=FZx(i,1);
    xsamm=Li/Nmitmeks; % element is divided into 4 parts
    xx=0;
    AP=AlgPar(i,:)';
#----- Transfer matrix equation -----
    for ij=1:Nmitmeks+1 # 5 - displacements and forces at x=0.0
        vvF=yflhlin(1.0,xx,EA,GAR,EI);
        vvB=yzhqz(1.0,xx,qx,qz,EA,EI);
        vvFz=yzfv(1.0,xx,aLx,Fx,Fz,EA,EI);
        Fvv(:,ij)=vvF*AP+vvB+vvFz;
        xx=xx+xsamm;
    endfor
    ...
endfor
#-----
```

2.8 The truss element

Let us now apply the local coordinate system (x, z) to the truss element shown in Fig. 2.20 (X, Z is the global coordinate system).

For this frame element, the rigid body translations and rigid body rotation at the ends of the element of Eq. (1.1) are shown in Fig.1.1.

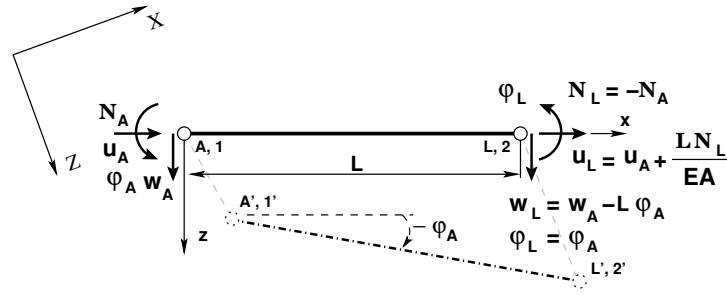


Figure 2.20. Truss element

The truss element displacements are described in local coordinates in Fig. 2.20.

$$u_L = u_A + \Delta u = u_A + \frac{LN}{EA} \quad (2.54)$$

$$w_L = w_A - L\varphi_A \quad (2.55)$$

$$\varphi_L = \varphi_A \quad (2.56)$$

where

u_A, u_L – axial displacements at the beginning and at the end of the element, respectively;
 w_A, w_L – transverse displacements at the beginning and at the end of the element, respectively;

φ_A, φ_L – rotation at the beginning and at the end of the element, respectively (free degrees of freedom);

$\Delta u = LN/EA$ – axial deformation of the element (internal displacement [KW90]),

EA – axial stiffness of the element;

L – length of the element.

Rigid body displacements and rotations of an element can be large. The axial deformation of an element is described in the basic coordinate system (in the co-rotational coordinate system [Yaw09]).

The truss element equilibrium equation:

$$N_L = -N_A \quad (2.57)$$

The basic equations of the truss element:

$$\mathbf{I}_{4 \times 4} \cdot \mathbf{Z}_L - \mathbf{U} \cdot \mathbf{Z}_A = 0 \quad (2.58)$$

where $\mathbf{Z}_L, \mathbf{Z}_A$ are the displacements and contact forces at the beginning and at the end of the element, respectively,

$$\mathbf{Z}_L = \begin{bmatrix} u_L \\ w_L \\ \varphi_L \\ \dots \\ N_L \end{bmatrix} \quad \mathbf{Z}_A = \begin{bmatrix} u_A \\ w_A \\ \varphi_A \\ \dots \\ N_A \end{bmatrix} \quad (2.59)$$

and the transfer matrix is given as

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & \vdots & -i_0 \cdot x/EA \\ 0 & 1 & -x & \vdots & 0 \\ 0 & 0 & 1 & \vdots & 0 \\ \dots & \dots & \dots & \vdots & \dots \\ 0 & 0 & 0 & \vdots & -1 \end{bmatrix} \quad (2.60)$$

Here, $i_0 = 1$ is the scaling multiplier for the displacements.

Let us rewrite the basic equations of the truss element of Eq. (2.58) in the following form:

$$\widehat{\mathbf{U}}_{4 \times 8} \cdot \widehat{\mathbf{Z}} = 0 \quad (2.61)$$

where

$$\widehat{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_L \\ \mathbf{Z}_A \end{bmatrix} \quad (2.62)$$

Equations (2.61) are inserted into the system of Eq. (2.1) with the GNU Octave function `ySpSRmhvI.m` (p. 245). This function uses the GNU Octave function `ySpSRhlin.m` (p. 245).

In excerpt 2.14 from the computing diary, GNU Octave commands for compiling the EST method equations can be found.

Computing diary excerpt 2.14 (`spESTframe93LaheWFI.m`)

```
#####
----- Sparse matrix instantiation -----
  spA=sparse(NNK,NNK)
spA =

Compressed Column Sparse (rows = 67, cols = 67, nnz = 0 [0%])

----- Right-hand side of the equations (RHS). -----
  B=zeros(NNK,1);

#####
---- Writing basic equations of frame ----
#####
-----The basic equations are inserted into spA -----
rows = rows_of_basic_equations: 30
col = cols_of_basic_equations: 60
spA_nnz = non_zero_elements_in_the_basic_equations: 95
```

```

#=====
Compatibility equations of displacements at nodes
From_row = Compatibility equations begin from row: 31
#=====
Node = 1
Node = 2
cmd = spA=spInsertBtoA(spA,31,19,SpTM3x3);
cmd = spA=spInsertBtoA(spA,31,1,SpTM3x3m);
Node = 3
Node = 4
cmd = spA=spInsertBtoA(spA,34,43,SpTM3x3);
cmd = spA=spInsertBtoA(spA,34,25,SpTM3x3m);
cmd = spA=spInsertBtoA(spA,37,43,SpTM3x3);
cmd = spA=spInsertBtoA(spA,37,13,SpTM3x3m);
Node = 5
Node = 6
cmd = spA=spInsertBtoA(spA,40,55,SpTM3x3);
cmd = spA=spInsertBtoA(spA,40,37,SpTM3x3m);
-----

spA_rows = 42
spA_cols = 60
spA_nnz = non_zero_elements_in_spA: 129

Compatibility equations of displacements are inserted into spA

compatibility_equations_rows = 12
non_zero_elements_in_compatibility_equations = 34

#=====
Joint equilibrium equations at nodes
From_rows = Joint equilibrium equations begin from row: 43
#=====
-----
Node = 1
-----
Number_of_reactions = The number of reactions at the node (node_no, reactions): 1, 2
cmd = spA=spInsertBtoA(spA,43,10,SpTM2x2);
cmd = spA=spInsertBtoA(spA,43,61,SpTM2x2xz);
-----
Node = 2
-----
cmd = spA=spInsertBtoA(spA,45,22,SpTM3x3);
cmd = spA=spInsertBtoA(spA,45,4,SpTM3x3);
cmd = B(45:47,1)=sF(1:3,1);
-----
Node = 3
-----
Number_of_reactions = The number of reactions at the node (node_no, reactions): 3, 3
cmd = spA=spInsertBtoA(spA,48,34,SpTM3x3);
cmd = spA=spInsertBtoA(spA,48,63,SpTM3x3xz);
-----

```

```

Node = 4
-----
cmd = spA=spInsertBtoA(spA,51,46,SpTM3x3);
cmd = spA=spInsertBtoA(spA,51,28,SpTM3x3);
cmd = spA=spInsertBtoA(spA,51,16,SpTM3x3);
cmd = B(51:53,1)=sF(1:3,1);
-----

Node = 5
-----
Number_of_reactions = The number of reactions at the node (node_no, reactions): 5, 2
cmd = spA=spInsertBtoA(spA,54,52,SpTM2x2);
cmd = spA=spInsertBtoA(spA,54,66,SpTM2x2xz);
-----

Node = 6
-----
cmd = spA=spInsertBtoA(spA,56,58,SpTM3x3);
cmd = spA=spInsertBtoA(spA,56,40,SpTM3x3);
cmd = B(56:58,1)=sF(1:3,1);
-----

spA_rows = 58
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 172

-----Equilibrium equations are inserted into spA -----

equilibrium_equations_rows = 16
non_zero_elements_in_equilibrium_equations = 43

#=====
Side conditions (hinges)
From_rows = Side conditions begin from row: 59
#=====
cmd = spA=spSisestaArv(spA,59,12,1);
cmd = spA=spSisestaArv(spA,60,54,1);
-----

spA_rows = 60
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 174

-----Side conditions are inserted into spA -----

side_condition_rows = 2
non_zero_elements_in_side_condition = 2

#=====
Restrictions on support displacements
From_rows = Restrictions on support displacements begin from row: 61
#=====

cmd = spA=spInsertBtoA(spA,61,7,SpTUv);

```

```

Support shift  tSiire(1,1,1) = 0  in x direction at node 1
cmd = B(61,1)=tSiire(1,1,1);

cmd = spA=spInsertBtoA(spA,62,7,SpTWv);
Support shift  tSiire(2,1,1) = 0  in z direction at node 1
cmd = B(62,1)=tSiire(2,1,1);

cmd = spA=spInsertBtoA(spA,63,31,SpTUv);
Support shift  tSiire(1,1,3) = 0  in x direction at node 3
cmd = B(63,1)=tSiire(1,1,3);

cmd = spA=spInsertBtoA(spA,64,31,SpTWv);
Support shift  tSiire(2,1,3) = 0  in z direction at node 3
cmd = B(64,1)=tSiire(2,1,3);

cmd = spA=spInsertBtoA(spA,65,31,SpTFiV);
Support shift  tSiire(3,1,3) = 0  in y direction at node 3
cmd = B(65,1)=tSiire(3,1,3);

cmd = spA=spInsertBtoA(spA,66,49,SpTUv);
Support shift  tSiire(1,1,5) = 0  in x direction at node 5
cmd = B(66,1)=tSiire(1,1,5);

cmd = spA=spInsertBtoA(spA,67,49,SpTWv);
Support shift  tSiire(2,1,5) = 0  in z direction at node 5
cmd = B(67,1)=tSiire(2,1,5);

-----

spA_rows = 67
spA_cols = 67
spA_nnz = non_zero_elements_in_spA: 181

-----Restrictions equations are inserted into spA -----

restrictions_equations_rows = 7
non_zero_elements_in_restrictions_equations = 7

#=====
spA_rank = 67
spA =

Compressed Column Sparse (rows = 67, cols = 67, nnz = 181 [4%])

```

3. Statically indeterminate problems

3.1 Illustrative frame problem 1

Example 3.1. Problem Statement. Depicted in Fig. 3.1 is a two-span frame. The height of the frame $h = 4\text{ m}$, the two spans are of the same length: $l = 6\text{ m}$. Let us assume that the flexural rigidity of the column $EI_p = 2 \cdot 10^4\text{ kN}\cdot\text{m}^2$ and that of the beam $EI_r = 2.0EI_p$, the axial rigidity of the column $EA_p = 4.6 \cdot 10^6\text{ kN}$ and that of the beam $EA_r = 8.8 \cdot 10^6\text{ kN}$, and the shear rigidity of the column $GA_{rp} = 0.4EA_p$ and that of the beam $GA_{rr} = 0.4EA_r$. A uniform load for the first span $q = 8\text{ kN/m}$. The central column is loaded with a concentrated load $F = 10\text{ kN}$.

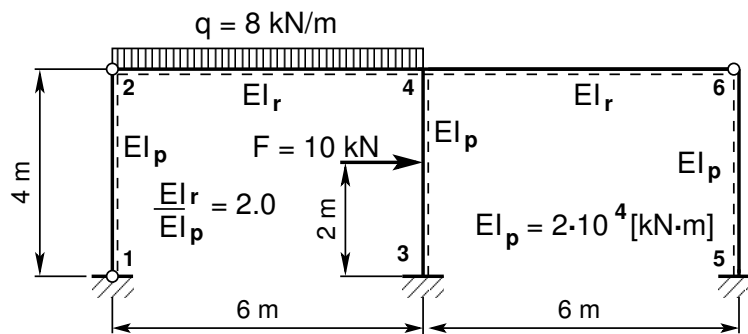


Figure 3.1. The two-span frame

We wish to compute the displacements, reactions, internal forces, and draw the axial force, shear force and bending moment diagrams.

Problem Solving. To solve the problem, we use the EST method. The solving procedure includes the following.

1. Data input: the number of frame nodes, elements, support reactions; element properties, elements loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, restrictions on support displacements.

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (3.1)$$

2. Assembling and solving the boundary problem equations (2.1) and (3.1) (prepared and solved by the program). To implement this aim, the program

(a) inserts the basic equations of a frame into the equation system,

- (b) adds the compatibility equations of the displacements at nodes to the equation system,
- (c) adds the joint equilibrium equations,
- (d) adds the side conditions (hinges),
- (e) adds the restrictions on support displacements,
- (f) solves the compiled system of sparse equations,
- (g) produces an output: initial parameter vectors for element displacements and forces; support reactions.

3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTframeLaheWFI.m* are shown in excerpts from the program: element and nodal loads – excerpt 3.1; nodal coordinates – excerpt 3.2; element properties, topology and hinges – excerpt 3.3.

Program excerpt 3.1 (spESTframeLaheWFI.m)

```

Number_of_frame_nodes=6
Number_of_elements=5
Number_of_support_reactions=8
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts ("Nmitmeks") of the element
Nmitmeks=4
# --- Element properties ---

EIp=20000 # kN/m^2           EAr=6.8*10^6
EIr=40000 # kN/m^2         #EAr=6.8*10^15;
EAp=4.6*10^6               GAp=0.4*EAp
#EAp=4.6*10^15;           GAR=0.4*EAr

baasi0=EIp/4 # scaling multiplier for displacements
# baasi0=1.0;
#Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];   esQkoormus(1,1:4,4)=[0.0 0.0 0.0 6.0];
esQkoormus(1,1:4,2)=[8.0 0.0 0.0 6.0];   esQkoormus(1,1:4,5)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

```



```

esFjoud(1,1:3,1)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 6.0];
esFjoud(1,1:3,3)=[10.0 0.0 2.0];

esFjoud(1,1:3,4)=[0.0 0.0 6.0];
esFjoud(1,1:3,5)=[0.0 0.0 4.0];

#
#Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0
#sSolmF(:,1,2)= 0.0
#sSolmF(:,1,3)= 0.0

#sSolmF(:,1,4)= 0.0
#sSolmF(:,1,5)= 0.0
#sSolmF(:,1,6)= 0.0

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0
#tSiire(2,1,1)= 0.01*baasi0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

#tSiire(:,1,4)= 0.0
#tSiire(:,1,5)= 0.0

```

Program excerpt 3.2 (Support reactions of the `spESTframeLaheWFI.m`)

```

#=====
#      Nodal coordinates
#=====
krdn=[#   x       z
      0.0    0.0;   % node 1
      0.0   -4.0;   % node 2
      6.0    0.0;   % node 3
      6.0   -4.0;   % node 4
     12.0    0.0;   % node 5
     12.0   -4.0]; % node 6
#=====
#
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No   u   w   fi
#=====
tsolm=[1     1 1 0;   % node 1
       3     1 1 1;   % node 3
       5     1 1 1]; % node 5
#=====

```

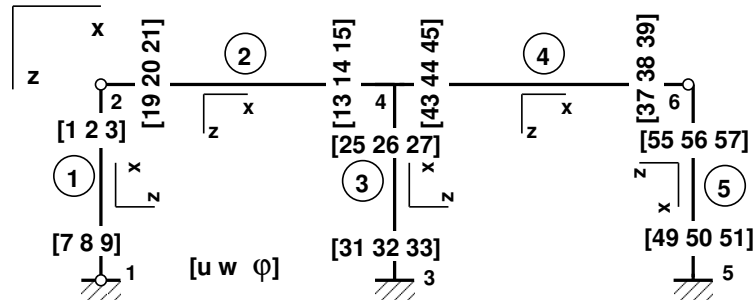


Figure 3.2. Numeration of displacements at frame element ends

Program excerpt 3.3 (`spESTframeLaheWFI.m`)

```
# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#         n2 - end of the element
#         n1 - beginning of the element
#         N, Q, M - hinges at the end of the element
#         N, Q, M - hinges at the beginning of the element
#
EIp EAp GAp  2   1   0 0 1   0 0 1;   % element 1
EIr EAr GAR  4   2   0 0 0   0 0 1;   % element 2
EIp EAp GAp  4   3   0 0 0   0 0 0;   % element 3
EIr EAr GAR  6   4   0 0 1   0 0 0;   % element 4
EIp EAp GAp  5   6   0 0 0   0 0 1]; % element 5
# 1 - hinge 'true' (axial, shear, moment hinges)
#
```

2. *Assembling and solving the boundary problem equations (3.1), carried out by the function `LaheFrameDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements at element ends of the frame as shown in Fig. 3.2 and forces at element ends of the frame as shown in Fig. 3.3.*

The unscaled initial parameter vectors of the elements are shown in excerpt 3.1 from the computing diary.

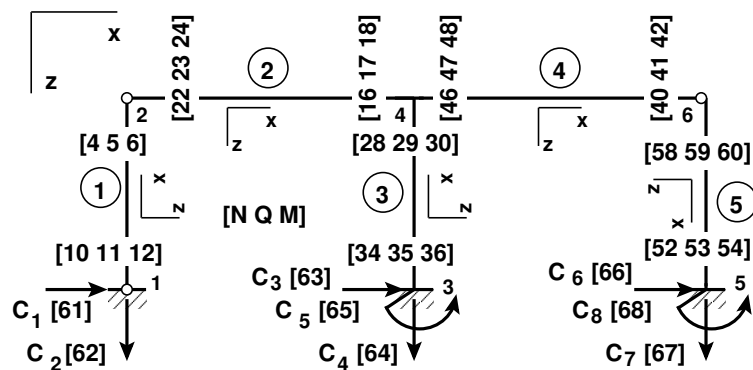


Figure 3.3. Numeration of forces at frame element ends

Computing diary excerpt 3.1 (spESTframeLaheWFI.m)

----- Scaling multiplier for displacements = 1/baasi0 -----

```
=====
Unscaled initial parameter vector
Element      u          w          fi          N          Q          M
No
-----
  1  -0.000e+00  0.000e+00  8.881e-06  20.301  0.000  0.000
  2  -3.552e-05  1.765e-05 -1.247e-03  0.000 -20.301  0.000
  3  -0.000e+00  0.000e+00  0.000e+00  29.980 -10.033  11.622
  4  -3.552e-05  2.607e-05  6.889e-04  -0.033 -2.281  13.684
  5  -1.983e-06  3.549e-05  1.331e-05  -2.281  0.033  0.000
-----
```

The support reactions of the frame in global coordinates are shown in excerpt 3.2 from the computing diary.

Computing diary excerpt 3.2 (spESTframeLaheWFI.m)

Support reactions begin from row X: 61

```
=====
No          X          Node    Cx <=> 1
          Cz <=> 2
          Cy <=> 3
-----
61  -0.000000e+00  1          1
62  -2.030089e+01  1          2
63  -1.003327e+01  3          1
64  -2.997974e+01  3          2
65  +1.162223e+01  3          3
66  +3.327478e-02  5          1
67  +2.280630e+00  5          2
68  -1.330991e-01  5          3
-----
```

3. Output: the element displacements and forces determined by the transfer matrix.

Computing diary excerpt 3.3 (spESTframeLaheWFI.m)

```
=====
Element displacements and forces determined by transfer matrix
#-----
Displacements and forces of element no 1 of length 4.000 m
The element is divided into 4 parts
displacement u - 0.000000e+00 -4.41324e-06 -8.82647e-06 -1.32397e-05 -1.76529e-05
displacement w - 0.000000e+00 -8.88062e-06 -1.77612e-05 -2.66418e-05 -3.55225e-05
rotation fi - 8.88062e-06 8.88062e-06 8.88062e-06 8.88062e-06 8.88062e-06
normal force N - -20.30089 -20.30089 -20.30089 -20.30089 -20.30089
shear force Q - 0.00000 0.00000 0.00000 0.00000 0.00000
```

moment force M - 0.00000 0.00000 0.00000 0.00000 0.00000

Displacements and forces of element no 2 of length 6.000 m

The element is divided into 4 parts

displacement u - -3.55225e-05 -3.55225e-05 -3.55225e-05 -3.55225e-05 -3.55225e-05
displacement w - 1.76529e-05 2.12539e-01 3.37647e+00 1.70839e+01 5.39892e+01
rotation fi - -1.24654e-03 -5.63176e-01 -4.49896e+00 -1.51836e+01 -3.59921e+01
normal force N - 0.00000 0.00000 0.00000 0.00000 0.00000
shear force Q - 20.30089 8.30089 -3.69911 -15.69911 -27.69911
moment force M - 0.00000 21.45134 24.90267 10.35401 -22.19465

Displacements and forces of element no 3 of length 4.000 m

The element is divided into 4 parts

displacement u - 0.00000e+00 -6.51733e-06 -1.30347e-05 -1.95520e-05 -2.60693e-05
displacement w - 0.00000e+00 2.06945e-04 4.93338e-04 4.17024e-01 3.33263e+00
rotation fi - 0.00000e+00 -3.30279e-04 -1.58895e-04 -1.24949e+00 -4.99831e+00
normal force N - -29.97974 -29.97974 -29.97974 -29.97974 -29.97974
shear force Q - 10.03327 10.03327 0.03327 0.03327 0.03327
moment force M - -11.62223 -1.58895 8.44432 8.47760 8.51087

Displacements and forces of element no 4 of length 6.000 m

The element is divided into 4 parts

displacement u - -3.55225e-05 -3.55151e-05 -3.55078e-05 -3.55004e-05 -3.54931e-05
displacement w - 2.60693e-05 -6.54442e-04 -7.57670e-04 -4.76041e-04 -1.98316e-06
rotation fi - 6.88865e-04 2.39865e-04 -8.08482e-05 -2.73276e-04 -3.37419e-04
normal force N - 0.03327 0.03327 0.03327 0.03327 0.03327
shear force Q - 2.28063 2.28063 2.28063 2.28063 2.28063
moment force M - -13.68378 -10.26284 -6.84189 -3.42095 0.00000

Displacements and forces of element no 5 of length 4.000 m

The element is divided into 4 parts

displacement u - -1.98316e-06 -1.48737e-06 -9.91578e-07 -4.95789e-07 4.23516e-22
displacement w - 3.54931e-05 2.24605e-05 1.10916e-05 3.05019e-06 3.38813e-21
rotation fi - 1.33099e-05 1.24780e-05 9.98244e-06 5.82309e-06 -1.69407e-21
normal force N - 2.28063 2.28063 2.28063 2.28063 2.28063
shear force Q - -0.03327 -0.03327 -0.03327 -0.03327 -0.03327
moment force M - 0.00000 -0.03327 -0.06655 -0.09982 -0.13310

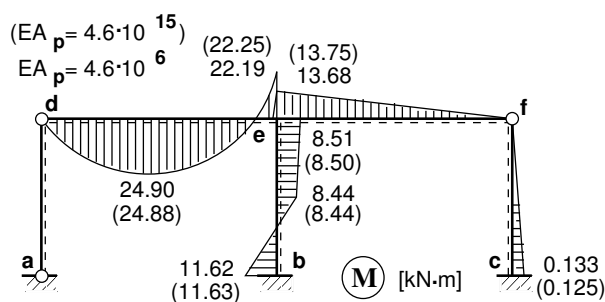


Figure 3.4. Bending moment diagram of the two-span frame

In Figs. 3.4 and 3.5, the values in brackets are determined with the axial stiffness $EA_p = 4.6 \cdot 10^{15}$ kN (with negligible axial deformations; the values coincide with these found with the force method, see [Lah12] Fig. 9.13 on p. 232).

The values without brackets are calculated with the axial stiffness $EA_p = 4.6 \cdot 10^6$ kN.

The shear force Q and the axial force N diagrams of the two-span frame are shown in Fig. 3.5.

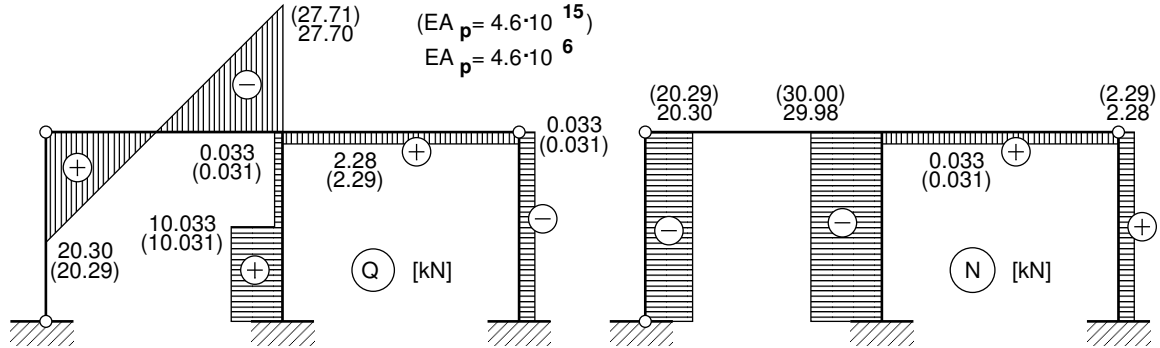


Figure 3.5. Diagrams of the axial force N and shear force Q of the two-span frame

Testing a static equilibrium for the frame

Consider a static equilibrium of the frame shown in Fig. 3.6. When projecting the forces onto the X-axis,

$$\Sigma X = 0; \quad 10.0 - 10.033 + 0.033 = 0 \tag{3.2}$$

and onto the Z-axis,

$$\Sigma Z = 0; \quad 8.0 \cdot 6.0 - 20.30 - 29.98 + 2.28 = 0 \tag{3.3}$$

We now write equation (3.4) of the sum of the moments and the moments of the forces acting about point a shown in Fig. 3.6:

$$\Sigma M_a = 0; \quad -8.0 \cdot 6.0 \cdot 3.0 - 10.0 \cdot 2.0 + 29.98 \cdot 6.0 + 11.62 - 2.28 \cdot 12.0 - 0.133 = 0.007 \text{ kN}\cdot\text{m} \approx 0 \tag{3.4}$$

The calculations with equations (3.2), (3.3), and (3.4) have verified the static equilibrium of the frame.

In Fig. 3.7, the elements, and in Fig. 3.8, the sparsity pattern of matrix spA of the two-span frame are shown.

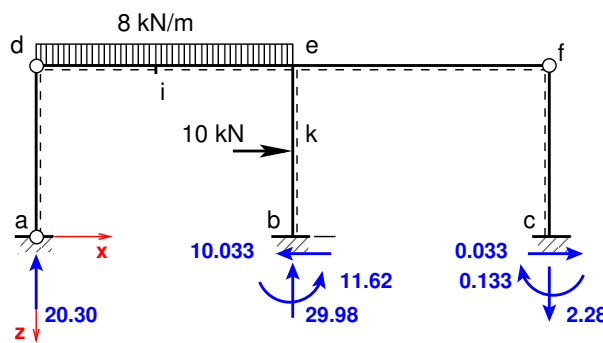


Figure 3.6. Boundary forces of the two-span frame

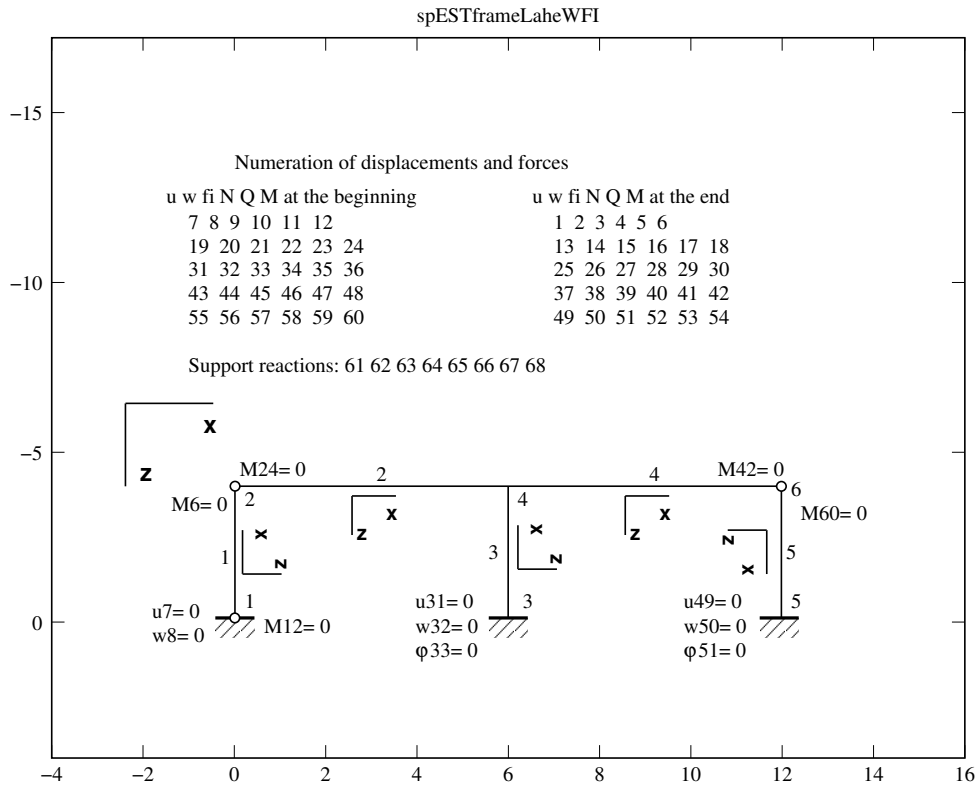


Figure 3.7. Elements of the two-span frame

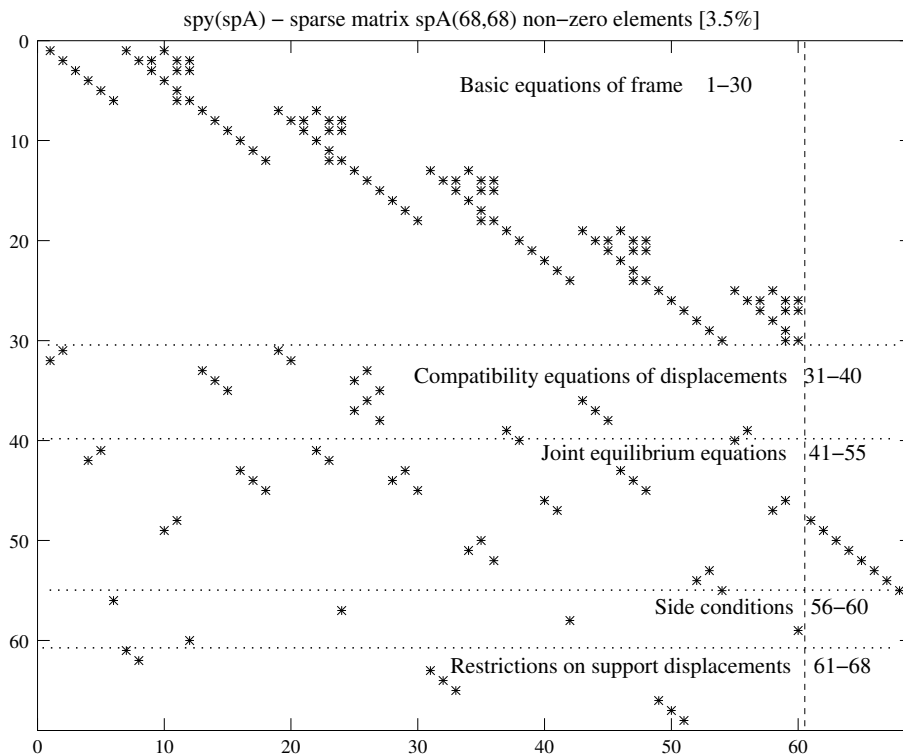


Figure 3.8. Sparsity pattern of matrix spA of the two-span frame

3.2 Illustrative frame problem 2

Example 3.2. Problem Statement. Depicted in Fig. 3.9 is a two-span frame. The height of the frame $h = 7$ m, the two spans are of the same length: $l = 9$ m. The rafter 2–4 is loaded with a vertical concentrated load $F = 16$ kN. The column 3–4 (of length $h_2 = 5.6$ m) is loaded with a uniform load $q = 16$ kN/m.

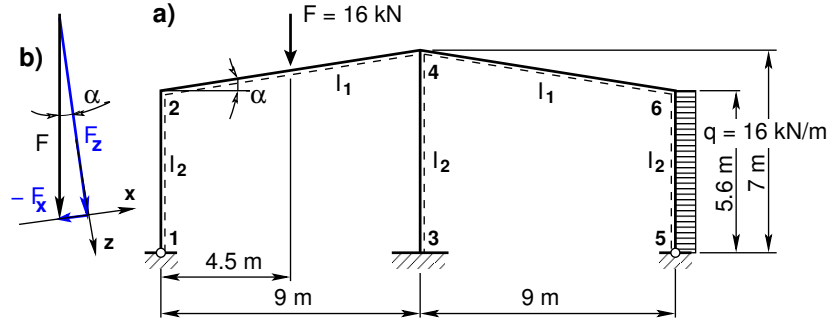


Figure 3.9. Two-span frame EST93

Assume that the value of the column flexural rigidity is $EI_p = 2 \cdot 10^4$ kN·m² and that of the rafter is $EI_r = 2.0EI_p$ ($I_1 = 2.0 \cdot I_2$); the axial rigidity of the column $EA_p = 4.6 \cdot 10^6$ kN and that of the rafter $EA_r = 8.8 \cdot 10^6$ kN; the shear rigidity of the column $GA_{rp} = 0.4EA_p$ and that of the rafter $GA_{rr} = 0.4EA_r$.

We wish to compute the displacements, reactions, internal forces, and draw the axial force, shear force and bending moment diagrams.

Problem Solving. To solve the problem, the EST method is used. Since the element loads are described in local coordinates, we assume the directions of the local coordinates shown in Fig. 3.10. Now we express the vertical concentrated load $F = 16$ kN (see Fig. 3.9 b) in local coordinates (F_x – projection onto the x -axis and F_z – projection onto the z -axis):

$$\begin{aligned} aF &= l_2/2, & \text{where } l_2 &= \sqrt{9^2 + 1.4^2} \\ F_x &= -F \cdot \sin \alpha, & \text{where } \sin \alpha &= 1.4/l_2 \\ F_z &= F \cdot \cos \alpha, & \text{where } \cos \alpha &= 9.0/l_2 \end{aligned} \quad (3.5)$$

Here, l_2 is the length of the rafter and $aF = l_2/2$ is the force F application point in local coordinates.

As in example 3.1, we carry out the following steps of calculations.

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (3.6)$$

2. Assembling and solving the boundary problem equations (2.1) and (3.6) (prepared and solved by the program). To implement this aim, the program

(a) inserts the basic equations of a frame into the equation system,

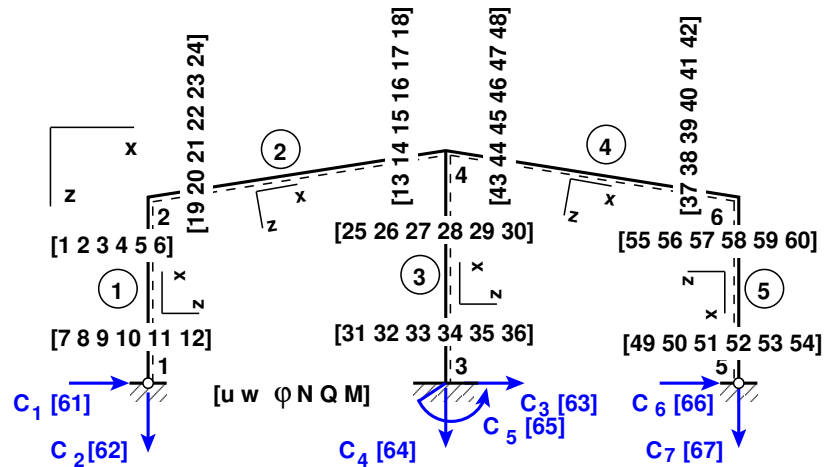


Figure 3.10. Numeration of displacements and forces of the two-span frame EST93

- (b) adds the compatibility equations of the displacements at nodes to the equation system,
- (c) adds the joint equilibrium equations,
- (d) adds the side conditions (hinges),
- (e) adds the restrictions on support displacements,
- (f) solves the compiled system of sparse equations,
- (g) produces an output: initial parameter vectors for element displacements and forces; support reactions.

3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTframe93LaheWFI.m* are shown in excerpts from the program: element and nodal loads – excerpt 3.4; nodal coordinates – excerpt 3.5; element properties, topology and hinges – excerpt 3.6.

Program excerpt 3.4 (spESTframe93LaheWFI.m)

```

Number_of_frame_nodes=6
Number_of_elements=5
Number_of_support_reactions=7
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces calculated on parts of the element 'Nmitmeks'.
Nmitmeks=4
Lp=18.0; # graphics axis
# --- Element properties ---

```



```

EIp=20000 # kN/m^2
EIr=40000 # kN/m^2
# EAp=4.6*10^6
EAp=4.6*10^15;
#EAR=6.8*10^6;
EAR=6.8*10^15
GAp=0.4*EAp;
GAR=0.4*EAR;

F2=16;
qz5=16;
L2x=sqrt(9^2+1.4^2); # length of the element
sinA2=1.4/L2x;
cosA2=9.0/L2x;
aF2=L2x/2;
Fz2=F2*cosA2;
Fx2=-F2*sinA2;

baasi0=EIp/5.6 # scaling multiplier for displacements
#baasi0=1.0;

#Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 5.6]; esQkoormus(1,1:4,4)=[0.0 0.0 0.0 L2x];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 L2x]; esQkoormus(1,1:4,5)=[qz5 0.0 0.0 5.6];
esQkoormus(1,1:4,3)=[0.0 0.0 0.0 7.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 5.6]; esFjoud(1,1:3,4)=[0.0 0.0 L2x];
esFjoud(1,1:3,2)=[Fz2 Fx2 aF2]; esFjoud(1,1:3,5)=[0.0 0.0 5.6];
esFjoud(1,1:3,3)=[0.0 0.0 7.0];

#
#Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0 #sSolmF(:,1,4)= 0.0
#sSolmF(:,1,2)= 0.0the #sSolmF(:,1,5)= 0.0
#sSolmF(:,1,3)= 0.0 #sSolmF(:,1,6)= 0.0

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

```

```
#tSiire(1,1,1)= 0.0
#tSiire(2,1,1)= 0.01*baasi0
#tSiire(1,1,3)= 0.0
#tSiire(2,1,3)= 0.0
#tSiire(3,1,3)= 0.0
#tSiire(1,1,5)= 0.0
#tSiire(2,1,5)= 0.0
```

Program excerpt 3.5 (`spESTframe93LaheWFI.m`)

```
#####
#      Nodal coordinates
#####
krdn=[#   x           z
      0.0      0.0;    # node 1
      0.0     -5.6;    # node 2
      9.0      0.0;    # node 3
      9.0     -7.0;    # node 4
      18.0     0.0;    # node 5
      18.0    -5.6];   # node 6
#####
#
#####
# Restrictions on support displacements (on - 1, off - 0)
# Support No   u   w   fi
#####
tsolm=[1      1 1 0;    % node 1
       3      1 1 1;    % node 3
       5      1 1 0];   % node 5
#####
```

Program excerpt 3.6 (`spESTframe93LaheWFI.m`)

```
# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#           n2 - end of the element
#           n1 - beginning of the element
#           N, Q, M - hinges at the end of the element
#           N, Q, M - hinges at the beginning of the element
#
EIp EAp GAp  2    1    0 0 0    0 0 1;    % element 1
EIr EAr GAr  4    2    0 0 0    0 0 0;    % element 2
EIp EAp GAp  4    3    0 0 0    0 0 0;    % element 3
EIr EAr GAr  6    4    0 0 0    0 0 0;    % element 4
EIp EAp GAp  5    6    0 0 1    0 0 0];   % element 5
# 1 - hinge 'true' (axial, shear, moment hinges)
#
```

2. *Assembling and solving the boundary problem equations (3.6), carried out by the function `LaheFrameDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the frame element ends as shown in Fig. 3.10. The unscaled initial parameter vectors of the elements are shown in excerpt 3.4 from the computing diary.*

Computing diary excerpt 3.4 (spESTframe93LaheWFI.m)

----- Scaling multiplier for displacements = 1/baasi0 -----

=====

Unscaled initial parameter vector						
Element No	u	w	fi	N	Q	M
1	-0.000e+00	0.000e+00	1.189e-02	22.670	12.850	0.000
2	-4.721e-02	-7.343e-03	1.815e-03	16.182	-20.426	71.959
3	-0.000e+00	0.000e+00	0.000e+00	-4.870	27.275	-102.642
4	-4.721e-02	7.343e-03	2.513e-03	39.371	-7.946	46.197
5	-2.192e-15	4.778e-02	2.340e-04	-1.800	-40.125	-26.181

The support reactions of the frame in global coordinates are shown in excerpt 3.5 from the computing diary.

Computing diary excerpt 3.5 (spESTframe93LaheWFI.m)

Support reactions begin from row X: 61

No	X	Node	Cx <=> 1 Cz <=> 2 Cy <=> 3
61	+1.284975e+01	1	1
62	-2.267047e+01	1	2
63	+2.727509e+01	3	1
64	+4.870101e+00	3	2
65	-1.026424e+02	3	3
66	+4.947516e+01	5	1
67	+1.800369e+00	5	2

3. **Output: the element displacements and forces** determined by the transfer matrix are shown in excerpt 3.6 from the computing diary.

The bending moment M , shear force Q and axial force N diagrams of the frame EST93 are shown in Fig. 3.11.

Computing diary excerpt 3.6 (spESTframe93LaheWFI.m)

=====

Element displacements and forces determined by transfer matrix

```
=====
Displacements and forces of element no 1 of length 5.600 m
The element is divided into 4 parts
displacement u - 0.00000e+00 -6.89971e-15 -1.37994e-14 -2.06991e-14 -2.75988e-14
displacement w - 0.00000e+00 -1.63514e-02 -3.09398e-02 -4.20022e-02 -4.77757e-02
rotation fi - 1.18894e-02 1.12598e-02 9.37089e-03 6.22270e-03 1.81524e-03
normal force N - -22.67047 -22.67047 -22.67047 -22.67047 -22.67047
shear force Q - -12.84975 -12.84975 -12.84975 -12.84975 -12.84975
moment force M - 0.00000 -17.98964 -35.97929 -53.96893 -71.95857
```

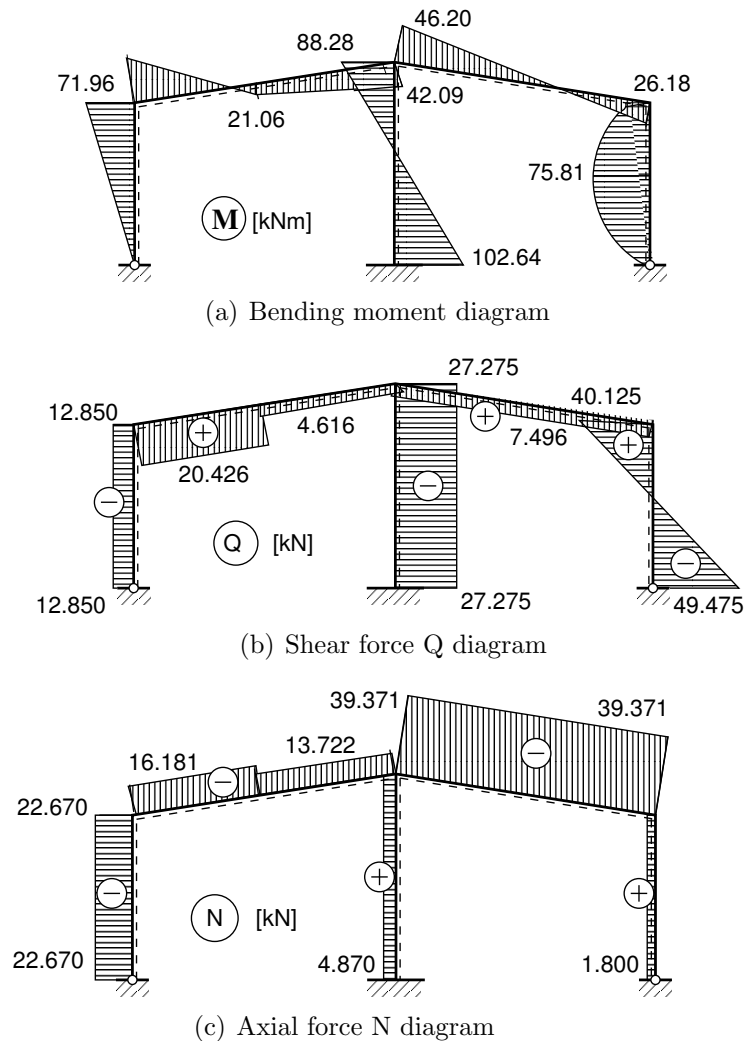


Figure 3.11. Internal forces diagrams of the two-span frame EST93

 Displacements and forces of element no 2 of length 9.108 m

The element is divided into 4 parts

displacement u	-4.72079e-02	-4.72079e-02	-4.72079e-02	-4.72079e-02	-4.72079e-02
displacement w	-7.34346e-03	-7.81788e-03	-4.99369e-03	2.77278e+00	2.22079e+01
rotation fi	1.81524e-03	-9.57252e-04	-1.08203e-03	-3.65812e+00	-1.46316e+01
normal force N	-16.18166	-16.18166	-13.72234	-13.72234	-13.72234
shear force Q	20.42597	20.42597	4.61611	4.61611	4.61611
moment force M	-71.95857	-25.44743	21.06372	31.57487	42.08602

Displacements and forces of element no 3 of length 7.000 m

The element is divided into 4 parts

displacement u	0.00000e+00	1.85276e-15	3.70551e-15	5.55827e-15	7.41102e-15
displacement w	0.00000e+00	-6.64042e-03	-2.16891e-02	-3.78371e-02	-4.77757e-02
rotation fi	0.00000e+00	6.89296e-03	9.60943e-03	8.14940e-03	2.51287e-03
normal force N	4.87010	4.87010	4.87010	4.87010	4.87010
shear force Q	-27.27509	-27.27509	-27.27509	-27.27509	-27.27509
moment force M	102.64244	54.91103	7.17962	-40.55179	-88.28320

 Displacements and forces of element no 4 of length 9.108 m

The element is divided into 4 parts

displacement u	-	-4.72079e-02	-4.72079e-02	-4.72079e-02	-4.72079e-02	-4.72079e-02
displacement w	-	7.34346e-03	4.22475e-03	4.74885e-03	6.57025e-03	7.34346e-03
rotation fi	-	2.51287e-03	3.98052e-04	-6.86705e-04	-7.41404e-04	2.33954e-04
normal force N	-	-39.37128	-39.37128	-39.37128	-39.37128	-39.37128
shear force Q	-	7.94644	7.94644	7.94644	7.94644	7.94644
moment force M	-	-46.19719	-28.10266	-10.00814	8.08639	26.18091

Displacements and forces of element no 5 of length 5.600 m

The element is divided into 4 parts

displacement u	-	-2.19175e-15	-1.64382e-15	-1.09588e-15	-5.47939e-16	0.00000e+00
displacement w	-	4.77757e-02	5.02581e-01	7.35198e+00	3.70545e+01	1.17045e+02
rotation fi	-	2.33954e-04	-1.30263e+00	-1.04416e+01	-3.52566e+01	-8.35876e+01
normal force N	-	1.80037	1.80037	1.80037	1.80037	1.80037
shear force Q	-	40.12484	17.72484	-4.67516	-27.07516	-49.47516
moment force M	-	26.18091	66.67568	75.81046	53.58523	-0.00000

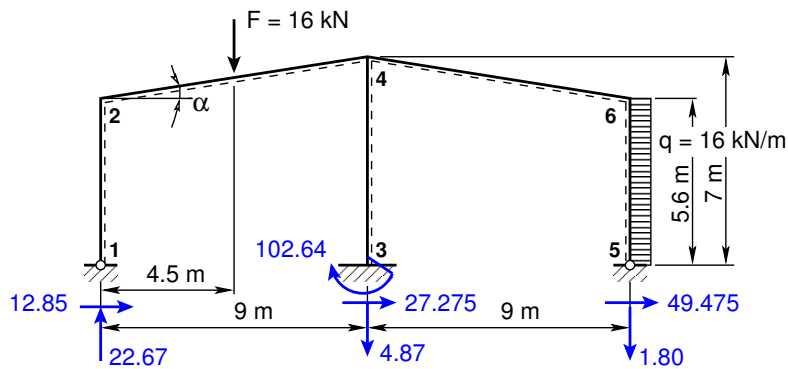


Figure 3.12. Boundary forces of the two-span frame EST93

Testing a static equilibrium for the frame

Consider a static equilibrium of the frame shown in Fig. 3.12. Let us project the forces onto the X-axis,

$$\Sigma X = 0; \quad 12.85 + 27.275 + 49.475 - 16 \cdot 5.6 = 0 \tag{3.7}$$

and onto the Z-axis,

$$\Sigma Z = 0; \quad 16.0 - 22.67 + 4.87 + 1.80 = 0 \tag{3.8}$$

The sum of the moments and the moments of the forces acting about point 1 shown in Fig. 3.12:

$$\begin{aligned} \Sigma M_1 = 0; \quad & -16.0 \cdot 4.5 - 4.8701 \cdot 9.0 - 102.64244 \\ & -1.800378 \cdot 18.0 + 16.0 \cdot 5.6 \cdot 2.8 = -1.44 \cdot 10^{-14} [kNm] \approx 0 \end{aligned} \tag{3.9}$$

The calculations with equations (3.7), (3.8), and (3.9) have verified the static equilibrium of the frame.

The elements and the sparsity pattern of matrix spA of the two-span frame are shown in Figs. 3.13 and 3.14, respectively.

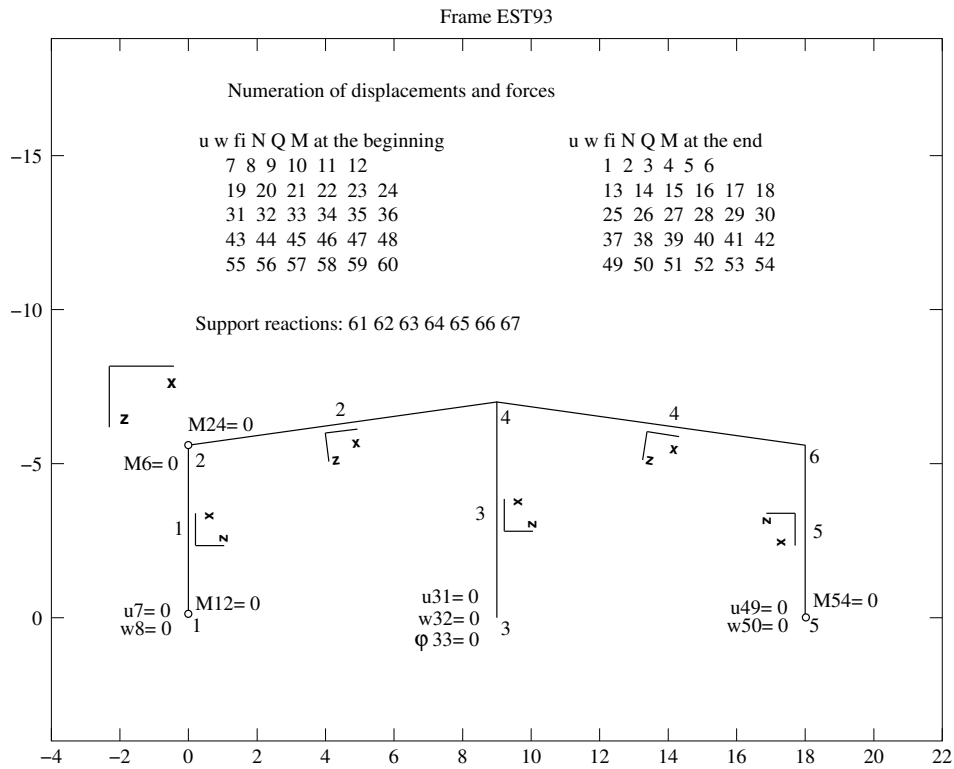


Figure 3.13. Elements of the two-span frame EST93

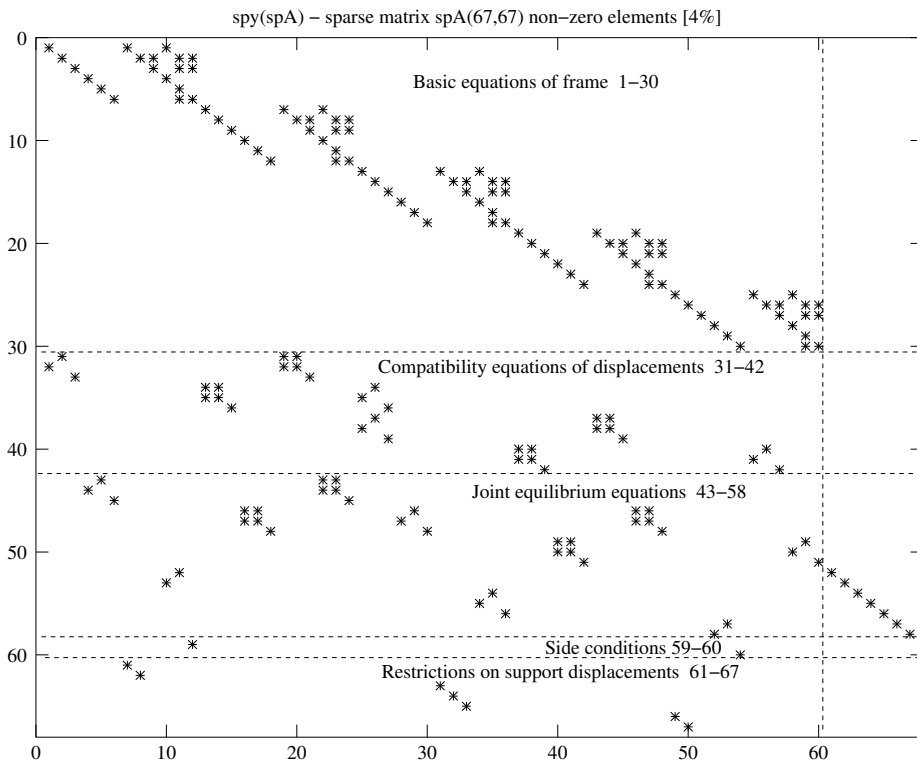


Figure 3.14. Sparsity pattern of matrix spA of the two-span frame EST93

3.3 Illustrative frame problem 3

Example 3.3. Problem Statement. The frame shown in Fig. 3.15 is of height $h = 7$ m and the two spans are of similar length: $l = 6$ m. The column 1–2 of length $h_2 = 5.0$ m is loaded with a uniform load $q_1 = 4$ kN/m. The rafter 2–4 is subjected to a vertical uniform load $q_2 = 8$ kN/m. The column 3–4 of length $h_2 = 7.0$ m carries a concentrated load $F = 12$ kN. The beam 3–5 is loaded with a vertical uniform load $q_3 = 8$ kN/m.

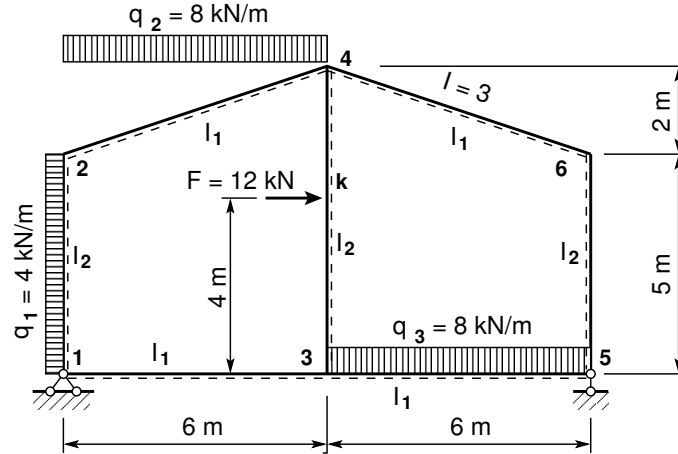


Figure 3.15. The frame EST77

Let us suppose that the value of the column flexural rigidity $EI_p = 2 \cdot 10^4$ kN·m² and the rafter flexural rigidity $EI_r = 3.0EI_p$ ($I_1 = 3.0 \cdot I_2$); the axial rigidity of the column $EA_p = 4.6 \cdot 10^6$ kN and that of the rafter $EA_r = 8.8 \cdot 10^6$ kN; the shear rigidity of the column $GA_{rp} = 0.4EA_p$ and that of the rafter $GA_{rr} = 0.4EA_r$.

We wish to compute the displacements, reactions, internal forces, and draw the axial force, shear force and bending moment diagrams.

Problem Solving. To solve the problem, the EST method is used. Since the element loads are described in local coordinates, we assume the directions of the local coordinates shown in Fig. 3.16. Now let us express the vertical uniform load, $q_2 = 8$ kN/m, in local coordinates (q_{2x} – projection onto the x -axis, q_{2z} – projection onto the z -axis):

$$\begin{aligned} q_{2k} &= q_2 \cdot 6.0/l_2, & \text{where } l_2 &= \sqrt{6^2 + 2.0^2} \\ q_{2x} &= -q_{2k} \cdot \sin \alpha, & \text{where } \sin \alpha &= 2.0/l_2 \\ q_{2z} &= q_{2k} \cdot \cos \alpha, & \text{where } \cos \alpha &= 6.0/l_2 \end{aligned} \quad (3.10)$$

Here, l_2 is the length of the rafter in local coordinates.

As in example 3.1, we carry out the following steps of calculations.

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (3.11)$$

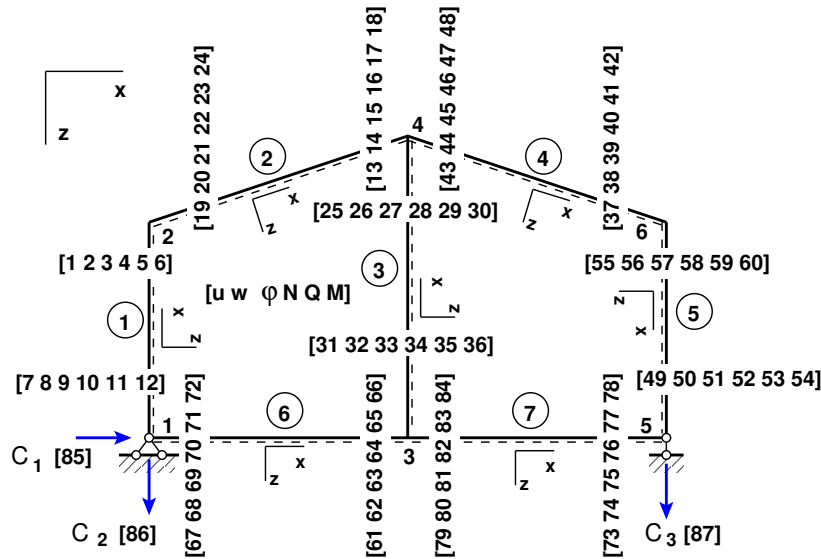


Figure 3.16. Numeration of displacements and forces of the frame EST77

2. *Assembling and solving the boundary problem equations (2.1) and (3.11) (prepared and solved by the program). To implement this aim, the program*

- (a) *inserts the basic equations of a frame into the equation system,*
- (b) *adds the compatibility equations of the displacements at nodes to the equation system,*
- (c) *adds the joint equilibrium equations,*
- (d) *adds the side conditions (hinges),*
- (e) *adds the restrictions on support displacements,*
- (f) *solves the compiled system of sparse equations,*
- (g) *produces an output: initial parameter vectors for element displacements and forces; support reactions.*

3. *Output: element displacements and forces determined by the transfer matrix.*

1. **Input data** for the GNU Octave program *spESTframe77LaheWFI.m* are shown in excerpts from the program: element and nodal loads – excerpt 3.7; nodal coordinates – excerpt 3.8; element properties, topology and hinges – excerpt 3.9.

Program excerpt 3.7 (spESTframe77LaheWFI.m)

```

Number_of_frame_nodes=6
Number_of_elements=7
Number_of_support_reactions=3
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces calculated on parts of the element 'Nmitmeks'.
Nmitmeks=4

```



```

Lp=12.0; # graphics axis
# --- Element properties ---
    EIp=20000 # kN/m^2          q2=8.0;
    EIr=40000 # kN/m^2          q3=8.0;
    # EAp=4.6*10^6              L2x=sqrt(6.0^2+2.0^2); # length of the element
    EAp=4.6*10^15;              sinA2=2.0/L2x;
    #EAr=6.8*10^6;              cosA2=6.0/L2x;
    EAr=6.8*10^15                qz2v=q2*6.0/L2x; # load/length of the element
    GAp=0.4*EAp;                qz1=q1;
    GAR=0.4*EAR;                qz7=q3;
    F3=12;                      qz2=qz2v*cosA2; # projection onto z-axis
    q1=4.0;                     qx2=-qz2v*sinA2; # projection onto x-axis

baasi0=EIp/5.0 # scaling multiplier for displacements
#baasi0=1.0;

#Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[qz1 0.0 0.0 5.0];      esQkoormus(1,1:4,5)=[0.0 0.0 0.0 5.0];
esQkoormus(1,1:4,2)=[qz2 qx2 0.0 L2x];      esQkoormus(1,1:4,6)=[0.0 0.0 0.0 6.0];
esQkoormus(1,1:4,3)=[0.0 0.0 0.0 7.0];      esQkoormus(1,1:4,7)=[qz7 0.0 0.0 6.0];
esQkoormus(1,1:4,4)=[0.0 0.0 0.0 L2x];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 0.0];      esFjoud(1,1:3,5)=[0.0 0.0 0.0];
esFjoud(1,1:3,2)=[0.0 0.0 0.0];      esFjoud(1,1:3,6)=[0.0 0.0 0.0];
esFjoud(1,1:3,3)=[12.0 0.0 4.0];      esFjoud(1,1:3,7)=[0.0 0.0 0.0];
esFjoud(1,1:3,4)=[0.0 0.0 0.0];

#
#Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0                  #sSolmF(:,1,4)= 0.0
#sSolmF(:,1,2)= 0.0                  #sSolmF(:,1,5)= 0.0
#sSolmF(:,1,3)= 0.0                  #sSolmF(:,1,6)= 0.0

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

```

```
#tSiire(1,1,1)= 0.0                                #tSiire(2,1,5)= 0.0
#tSiire(2,1,1)= 0.01*baasi0
```

Program excerpt 3.8 (`spESTframe77LaheWFI.m`)

```
#####
#      Nodal coordinates
#####
krdn=[#   x           z
      0.0    0.0;    # node 1
      0.0   -5.0;    # node 2
      6.0    0.0;    # node 3
      6.0   -7.0;    # node 4
     12.0    0.0;    # node 5
     12.0   -5.0];  # node 6
#####
#
#####
# Restrictions on support displacements (on - 1, off - 0)
# Support No   u   w   fi
#####
tsolm=[1     1 1 0;   % node 1
       5     0 1 0]; % node 5
#####
```

Program excerpt 3.9 (`spESTframe77LaheWFI.m`)

```
# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#           n2 - end of the element
#           n1 - beginning of the element
#           N, Q, M - hinges at the end of the element
#           N, Q, M - hinges at the beginning of the element
#
EIp EAp GAp  2    1    0 0 0    0 0 1;    % element 1
EIr EAr GAR  4    2    0 0 0    0 0 0;    % element 2
EIp EAp GAp  4    3    0 0 0    0 0 0;    % element 3
EIr EAr GAR  6    4    0 0 0    0 0 0;    % element 4
EIp EAp GAp  5    6    0 0 1    0 0 0;    % element 5
EIr EAr GAR  3    1    0 0 0    0 0 1;    % element 6
EIr EAr GAR  5    3    0 0 1    0 0 0];    % element 7
# 1 - hinge 'true' (axial, shear, moment hinges)
#
```

2. *Assembling and solving the boundary problem equations (3.11), carried out by the function `LaheFrameDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the frame element ends as shown in Fig. 3.16. The unscaled initial parameter vectors of the elements are shown in excerpt 3.7 from the computing diary.*

Computing diary excerpt 3.7 (spESTframe77LaheWFI.m)

----- Scaling multiplier for displacements = 1/baasi0 -----

```

=====
Unscaled initial parameter vector
Element  u          w          fi          N          Q          M
No
-----
  1 -0.000e+00  0.000e+00 -2.842e-03   37.423   -6.458    0.000
  2  1.204e-02  4.013e-03 -2.972e-03   24.682  -31.221   17.712
  3 -1.021e-02  2.254e-14 -1.220e-03  -12.982  -14.240   37.191
  4  1.850e-02  4.598e-03  5.078e-04   18.172   18.775  -62.236
  5  2.561e-14 -1.950e-02  8.096e-04   23.558  -11.302   56.510
  6  0.000e+00  0.000e+00 -1.943e-03  -25.542   -2.410    0.000
  7  2.254e-14  1.021e-02 -1.220e-03  -11.302  -15.392  -51.650
-----

```

The support reactions of the frame in global coordinates are shown in excerpt 3.8 from the computing diary.

Computing diary excerpt 3.8 (spESTframe77LaheWFI.m)

Support reactions begin from row X: 85

```

=====
No          X          Node    Cx <=> 1
          Cz <=> 2
          Cy <=> 3
-----
 85  -3.200000e+01    1         1
 86  -3.983333e+01    1         2
 87  -5.616667e+01    5         2
-----

```

3. Output: the element displacements and forces determined by the transfer matrix are shown in excerpt 3.9 from the computing diary. The diagrams of the bending moment M , shear force Q and axial force N of the frame EST77 are shown in Fig. 3.17.

Here, the values are determined with the axial stiffness $EA_p = 4.6 \cdot 10^{15}$ kN (with negligible axial deformations, the values coincide with these found with the force method, see [Lah12]¹ Fig. 9.23 on p. 243).

¹http://digi.lib.ttu.ee/opik_eme/Ehitusmehaanika.pdf#page=243.

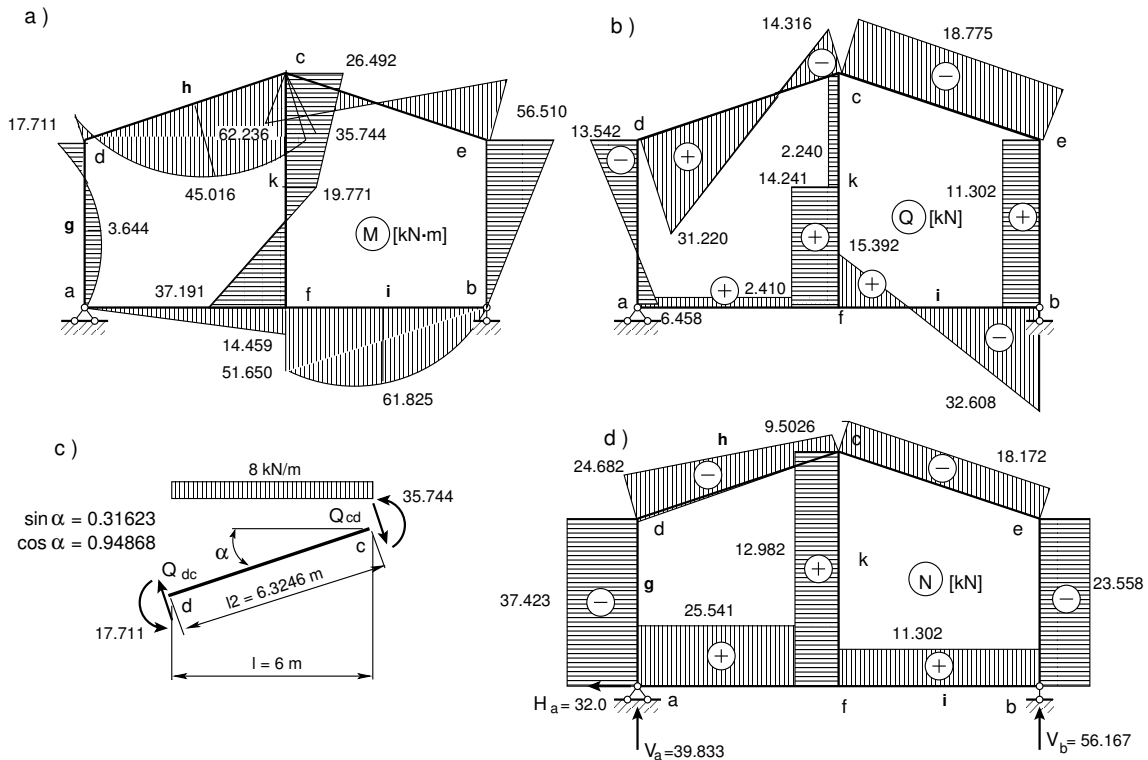


Figure 3.17. Internal forces diagrams of the frame EST77

Computing diary excerpt 3.9 (`spESTframe77LaheWFI.m`)

```

=====
Element displacements and forces determined by transfer matrix
=====
Displacements and forces of element no 1 of length 5.000 m
The element is divided into 4 parts
displacement u - 0.00000e+00 -1.01694e-14 -2.03388e-14 -3.05082e-14 -4.06777e-14
displacement w - 0.00000e+00 8.48273e-02 1.30835e+00 6.59962e+00 2.08408e+01
rotation fi - -2.84172e-03 -2.63006e-01 -2.08517e+00 -7.03182e+00 -1.66655e+01
normal force N - -37.42345 -37.42345 -37.42345 -37.42345 -37.42345
shear force Q - 6.45764 1.45764 -3.54236 -8.54236 -13.54236
moment force M - 0.00000 4.94705 3.64410 -3.90885 -17.71180
-----
Displacements and forces of element no 2 of length 6.32456 m
The element is divided into 4 parts
displacement u - 1.20390e-02 1.20390e-02 1.20390e-02 1.20390e-02 1.20390e-02
displacement w - 4.01301e-03 1.33739e-01 2.01215e+00 1.01372e+01 3.20068e+01
rotation fi - -2.97237e-03 -3.19016e-01 -2.53113e+00 -8.53667e+00 -2.02330e+01
normal force N - -24.68175 -20.88701 -17.09228 -13.29755 -9.50281
shear force Q - 31.22053 19.83633 8.45213 -2.93207 -14.31626
moment force M - -17.71180 22.65220 45.01619 49.38019 35.74419
-----
Displacements and forces of element no 3 of length 7.000 m
The element is divided into 4 parts

```

```

displacement u - -1.02109e-02 -1.02109e-02 -1.02109e-02 -1.02109e-02 -1.02109e-02
displacement w -  2.25374e-14  4.34616e-03  1.05712e-02  7.96109e-01  1.08134e+01
rotation   fi - -1.21985e-03 -3.38376e-03 -3.36712e-03 -1.87617e+00 -1.07968e+01
normal force N -  12.98177    12.98177    12.98177    12.98177    12.98177
shear force Q -  14.24039    14.24039    14.24039     2.24039     2.24039
moment force M - -37.19081    -12.27013    12.65055    22.57123    26.49192
-----

```

Displacements and forces of element no 4 of length 6.32456 m

The element is divided into 4 parts

```

displacement u -  1.84970e-02  1.84970e-02  1.84970e-02  1.84970e-02  1.84970e-02
displacement w -  4.59761e-03  2.70421e-03 -5.45432e-04 -3.91437e-03 -6.16566e-03
rotation   fi -  5.07846e-04  1.75676e-03  2.22336e-03  1.90766e-03  8.09646e-04
normal force N - -18.17178    -18.17178    -18.17178    -18.17178    -18.17178
shear force Q - -18.77538    -18.77538    -18.77538    -18.77538    -18.77538
moment force M -  62.23611     32.54962      2.86313     -26.82336    -56.50985
-----

```

Displacements and forces of element no 5 of length 5.000 m

The element is divided into 4 parts

```

displacement u -  2.56069e-14  1.92051e-14  1.28034e-14  6.40172e-15 -3.15544e-30
displacement w - -1.94975e-02 -1.84861e-02 -1.41636e-02 -7.63366e-03 -3.25087e-14
rotation   fi -  8.09646e-04 -2.28074e-03 -4.48815e-03 -5.81260e-03 -6.25409e-03
normal force N - -23.55832    -23.55832    -23.55832    -23.55832    -23.55832
shear force Q -  11.30197     11.30197     11.30197     11.30197     11.30197
moment force M - -56.50985    -42.38239    -28.25493    -14.12746     0.00000
-----

```

Displacements and forces of element no 6 of length 6.000 m

The element is divided into 4 parts

```

displacement u -  0.00000e+00  5.63434e-15  1.12687e-14  1.69030e-14  2.25374e-14
displacement w -  0.00000e+00  2.89162e-03  5.64769e-03  8.13265e-03  1.02109e-02
rotation   fi - -1.94281e-03 -1.89763e-03 -1.76207e-03 -1.53614e-03 -1.21985e-03
normal force N -  25.54236     25.54236     25.54236     25.54236     25.54236
shear force Q -   2.40988     2.40988     2.40988     2.40988     2.40988
moment force M -   0.00000     3.61482      7.22964     10.84447     14.45929
-----

```

Displacements and forces of element no 7 of length 6.000 m

The element is divided into 4 parts

```

displacement u -  2.25374e-14  2.50305e-14  2.75235e-14  3.00166e-14  3.25097e-14
displacement w -  1.02109e-02  1.23428e-01  1.80884e+00  9.11559e+00  2.87928e+01
rotation   fi - -1.21985e-03 -2.99640e-01 -2.39748e+00 -8.09475e+00 -1.91914e+01
normal force N -  11.30197     11.30197     11.30197     11.30197     11.30197
shear force Q -  15.39165     3.39165     -8.60835    -20.60835    -32.60835
moment force M -  51.65010     65.73758     61.82505     39.91253     0.00000
-----

```

Forces of element 3 at x = 4.00 m

```

displacement u - -1.02109e-02
displacement w -  1.21608e-02
rotation   fi - -2.96185e-03
normal force N -  12.98177
shear force Q -   2.24039
moment force M -  19.77075
-----

```

Forces of element 3 at x = 3.99 m

```

displacement u - -1.02109e-02
displacement w -  1.21608e-02
rotation   fi - -2.96185e-03
normal force N -  12.98177
shear force Q -  14.24039
moment force M -  19.77073
-----

```

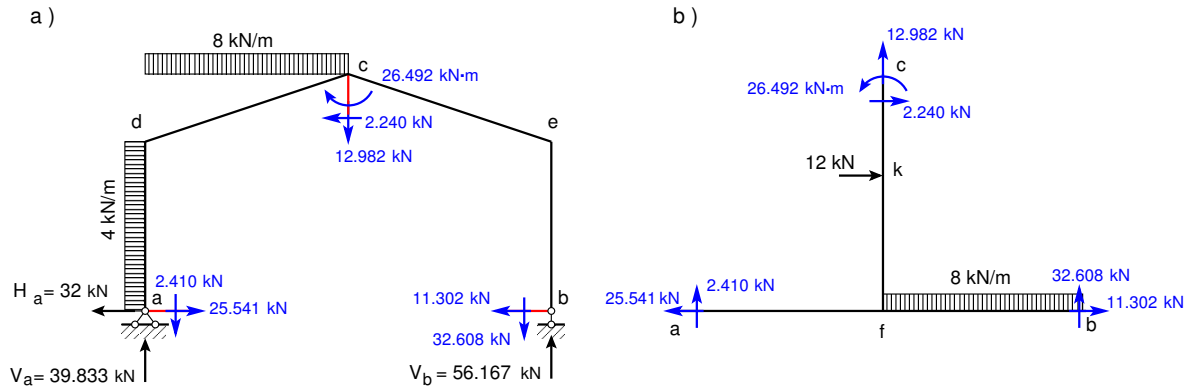


Figure 3.18. Boundary forces of the frame EST77

Testing a static equilibrium for the frame

Consider now a static equilibrium of the frame shown in Fig. 3.18 a. We project the forces onto the X and Z axes. The sum of the moments and the moments of the forces acting about point a (shown in Fig. 3.18 a) are:

$$\begin{aligned}
 \Sigma X &= 0; & -32.0 + 25.541 + 4 \cdot 5 - 2.240 - 32.608 &= -1.0 \cdot 10^{-3} \cong 0 \\
 \Sigma Y &= 0; & 39.833 - 2.410 - 8 \cdot 6 - 12.982 - 32.608 + 56.167 &= 7.1 \cdot 10^{-15} \cong 0 \\
 \Sigma M_a &= 0; & -4 \cdot 5 \cdot 2.5 - 8 \cdot 6 \cdot 3 - 12.982 \cdot 6 + 2.240 \cdot 7 \\
 & & + (-32.608 + 56.167) \cdot 12 - 26.492 &= 4.0 \cdot 10^{-3} \cong 0
 \end{aligned} \tag{3.12}$$

Consider next a static equilibrium of the frame shown in Fig. 3.18 b. Let us project the forces onto the X and Z axes. Now, the sum of the moments and the moments of the forces acting about point a (shown in Fig. 3.18 b) are:

$$\begin{aligned}
 \Sigma X &= 0; & -25.541 + 12 + 2.240 + 11.302 &= -1.0 \cdot 10^{-3} \cong 0 \\
 \Sigma Y &= 0; & +2.4103 + 12.982 - 8 \cdot 6 + 32.608 &= -7.1 \cdot 10^{-15} \cong 0 \\
 \Sigma M_a &= 0; & -2.240 \cdot 7 + 12.982 \cdot 6 + 26.492 - 12 \cdot 4 \\
 & & - 8 \cdot 6 \cdot 9 + 32.608 \cdot 12 &= -5.68 \cdot 10^{-14} \cong 0
 \end{aligned} \tag{3.13}$$

The calculations with Eqs. (3.12) and (3.13) have verified the static equilibrium of the frame.

The elements and the sparsity pattern of matrix spA of the frame EST77 are shown in Figs. 3.19 and 3.20, respectively.

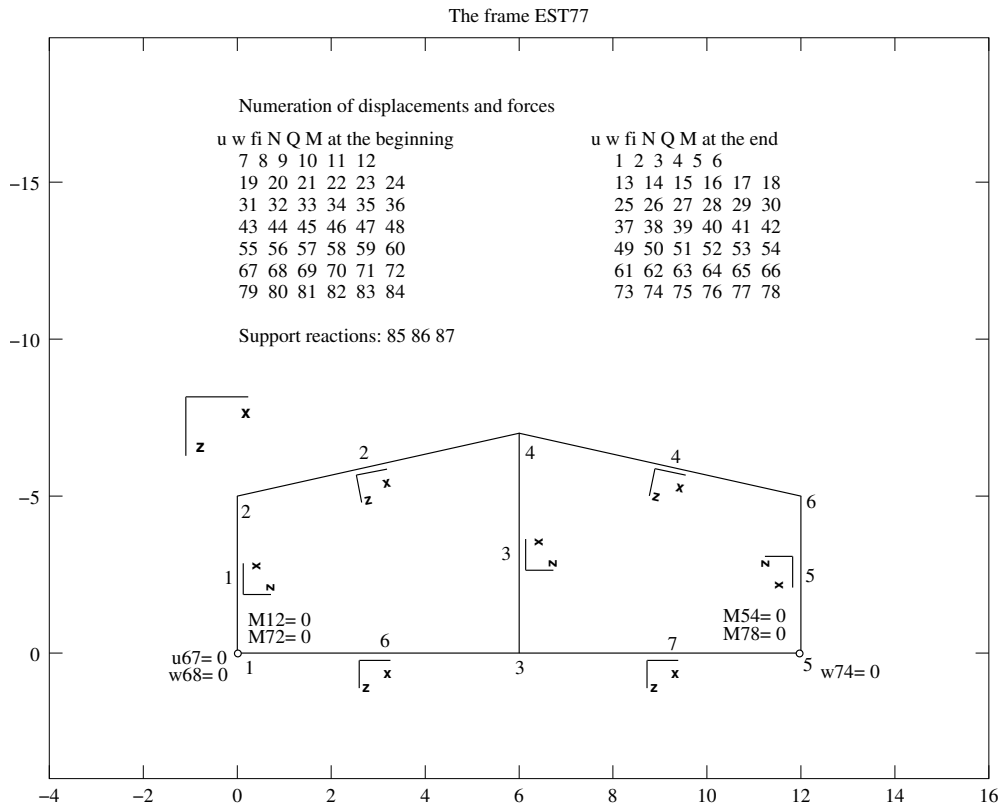


Figure 3.19. Elements of the frame EST77

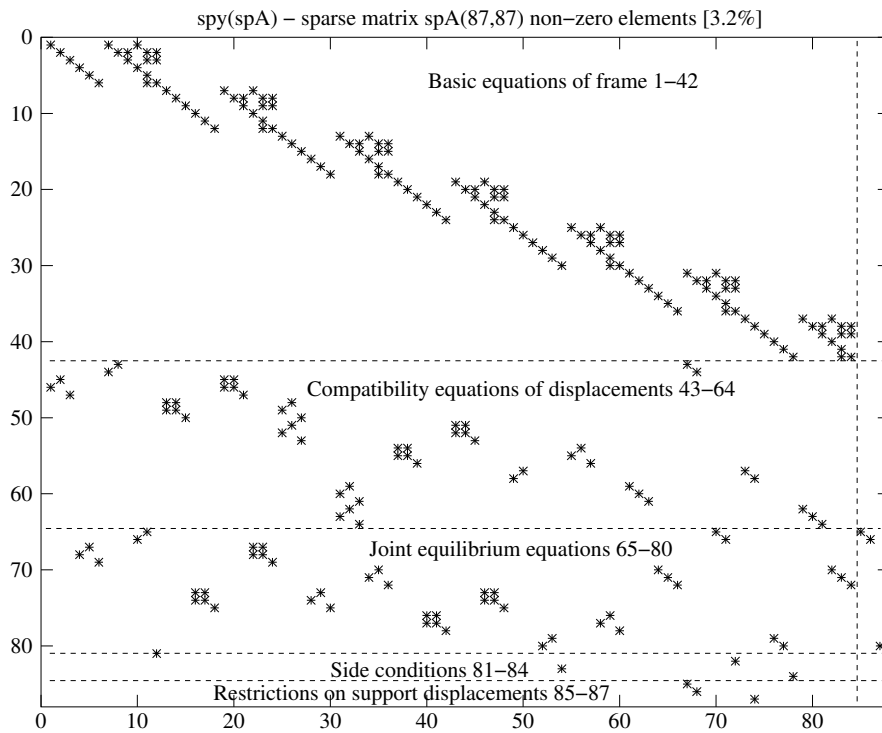


Figure 3.20. Sparsity pattern of matrix spA of the frame EST77

3.4 Illustrative continuous beam problem

Example 3.4. Problem Statement. Given the continuous beam depicted in Fig. 3.21, find the reactions, displacements and internal forces. Draw the shear force and bending moment diagrams. Assume that the value of the beam flexural rigidity $EI = 2 \cdot 10^4 \text{ kN}\cdot\text{m}^2$ and the shear rigidity $GA_r = 1.0 \cdot 10^{15} \text{ kN}$. The dead loading is a uniform load $q = 12 \text{ kN/m}$ (load case 1). The beam is also subjected to a live loading (load cases 2 and 3). The beam is simultaneously loaded with forces $F_1 = 60 \text{ kN}$ and $F_2 = 40 \text{ kN}$ (load case 2). In load case 3, it is loaded with force $F_3 = 80 \text{ kN}$.

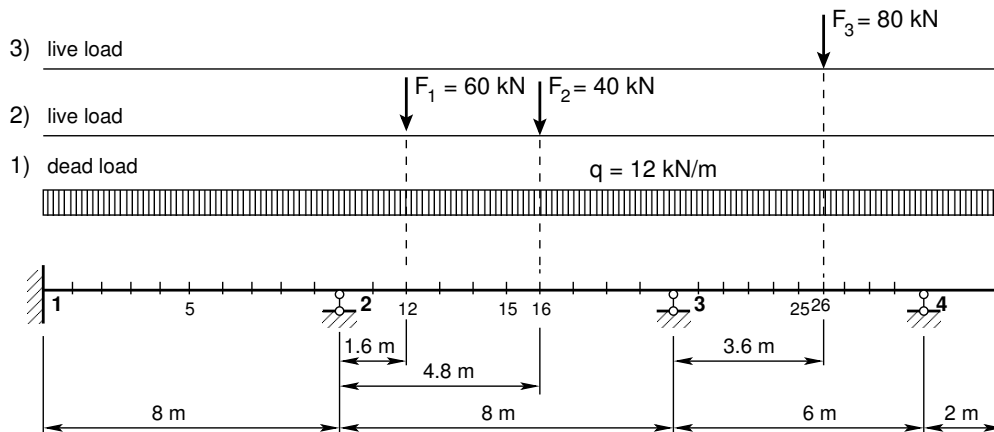


Figure 3.21. The continuous beam EST

Problem Solving. To solve the problem, we use the EST method. The solving procedure includes the following.

1. Data input: the number of beam nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, the topology and hinges, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (3.14)$$

2. Assembling and solving the boundary problem equations (2.1) and (3.14) (prepared and solved by the program). To implement this aim, the program
 - (a) inserts the basic equations of beam into the equation system,
 - (b) adds the compatibility equations of the displacements at nodes to the equation system,
 - (c) adds the joint equilibrium equations,
 - (d) adds the side conditions (hinges),
 - (e) adds the restrictions on support displacements,
 - (f) solves the compiled system of sparse equations,
 - (g) produces an output: initial parameter vectors for element displacements and forces; support reactions.

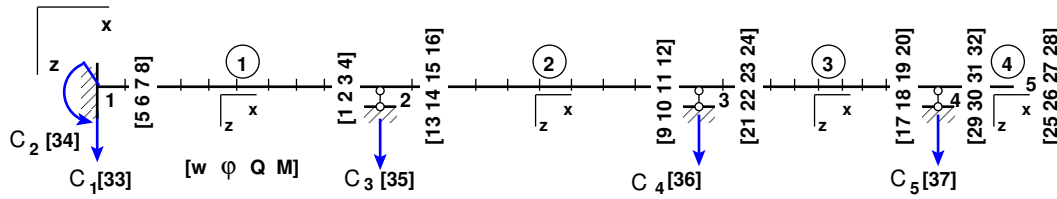


Figure 3.22. Numeration of displacements and forces of the continuous beam EST

3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTbeamLaheWFI.m* are shown in excerpts from the program: element and nodal loads – excerpt 3.10; nodal coordinates – excerpt 3.11; element properties, topology and hinges – excerpt 3.12.

Program excerpt 3.10 (spESTbeamLaheWFI.m)

```

Number_of_beam_nodes=5
Number_of_elements=4
Number_of_support_reactions=5
spNNK=8*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces calculated on parts of the element 'Nmitmeks'.
Nmitmeks=5
Lp=24.0; # graphics axis
# --- Element properties ---

EI=20000 # kN/m^2
GA=1.0*10^15
F1=60;
F2=40;
F3=80;
qz=12.0;

baasi0=EI/8.0 # scaling multiplier for displacements
#baasi0=1.0;

# ---- load variants -----

load_variant=1
#load_variant=2
#load_variant=3
koormusvariant=load_variant
#koormusvariant=2
#koormusvariant=3

switch (koormusvariant)
case{1}
#
disp(' Load variant 1 ')
#disp(' Element load in local coordinates ')
# Uniformly distributed load in local coordinate z direction
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,3,ElementideArv);

esQkoormus(1,1:3,1)=[qz 0.0 8.0];
esQkoormus(1,1:3,2)=[qz 0.0 8.0];
esQkoormus(1,1:3,3)=[qz 0.0 6.0];
esQkoormus(1,1:3,4)=[qz 0.0 2.0];

```

```

# Point load in local coordinate z direction
LoadsF_on_Element=5;
# 2 - Fz, aFz (coordinate of the point of force Fz application)
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:2,1)=[0.0 8.0];          esFjoud(1,1:2,3)=[0.0 6.0];
esFjoud(1,1:2,2)=[0.0 8.0];          esFjoud(1,1:2,4)=[0.0 2.0];
esFjoud(2,1:2,2)=[0.0 8.0];

#disp(' Node forces in global coordinates ')
#disp(' Node 1 load ')
# sSolmF(forces,1,nodes); forces=[Fz; My]
sSolmF = zeros(2,1,SolmedeArv);

#sSolmF(1,1,1)= 0.0 # force Fz          #sSolmF(:,1,3)= 0.0
#sSolmF(2,1,1)= 0.0 # force My          #sSolmF(:,1,4)= 0.0
#sSolmF(:,1,2)= 0.0                     #sSolmF(:,1,5)= 0.0

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(2,1,SolmedeArv);

#tSiire(:,1,1)= 0.0                     #tSiire(:,1,3)= 0.0
#tSiire(2,1,1)= 0.01*baasi0             #tSiire(:,1,4)= 0.0
#tSiire(:,1,2)= 0.0                     #tSiire(:,1,5)= 0.0

case{2}
disp(' Load variant 2 ')
#disp(' Element load in local coordinates ')
# Uniformly distributed load in local coordinate z direction
LoadsqONElement=4;
esQkoormus=zeros(LoadsqONElement,3,ElementideArv);

esQkoormus(1,1:3,1)=[0.0 0.0 8.0];      esQkoormus(1,1:3,3)=[0.0 0.0 6.0];
esQkoormus(1,1:3,2)=[0.0 0.0 8.0];      esQkoormus(1,1:3,4)=[0.0 0.0 2.0];

# Point load in local coordinate z direction
LoadsF_on_Element=5;
# 2 - Fz, aFz (coordinate of the point of force Fz application)
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:2,1)=[0.0 8.0];          esFjoud(1,1:2,3)=[0.0 6.0];
esFjoud(1,1:2,2)=[F1 1.6];          esFjoud(1,1:2,4)=[0.0 2.0];
esFjoud(2,1:2,2)=[F2 4.8];

#disp(' Node forces in global coordinates ')
#disp(' Node 1 load ')
# sSolmF(forces,1,nodes); forces=[Fz; My]
sSolmF = zeros(2,1,SolmedeArv);

```

```

#sSolmF(1,1,1)= 0.0 # force Fz
#sSolmF(2,1,1)= 0.0 # force My
#sSolmF(:,1,2)= 0.0

#sSolmF(:,1,3)= 0.0
#sSolmF(:,1,4)= 0.0
#sSolmF(:,1,5)= 0.0

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(2,1,SolmedeArv);

#tSiire(:,1,1)= 0.0
#tSiire(2,1,1)= 0.01*baasi0
#tSiire(:,1,2)= 0.0

#tSiire(:,1,3)= 0.0
#tSiire(:,1,4)= 0.0
#tSiire(:,1,5)= 0.0

case{3}
disp(' Load variant 3 ')
#disp(' Element load in local coordinates ')
# Uniformly distributed load in local coordinate z direction
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,3,ElementideArv);

esQkoormus(1,1:3,1)=[0.0 0.0 8.0];
esQkoormus(1,1:3,2)=[0.0 0.0 8.0];
esQkoormus(1,1:3,3)=[0.0 0.0 6.0];
esQkoormus(1,1:3,4)=[0.0 0.0 2.0];

# Point load in local coordinate z direction
LoadsF_on_Element=5;
# 2 - Fz, aFz (coordinate of the point of force Fz application)
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:2,1)=[0.0 8.0];
esFjoud(1,1:2,2)=[0.0 8.0];
esFjoud(2,1:2,2)=[0.0 8.0];
esFjoud(1,1:2,3)=[F3 3.6];
esFjoud(1,1:2,4)=[0.0 2.0];

#disp(' Node forces in global coordinates ')
#disp(' Node 1 load ')
# sSolmF(forces,1,nodes); forces=[Fz; My]
sSolmF = zeros(2,1,SolmedeArv);

#sSolmF(1,1,1)= 0.0 # force Fz
#sSolmF(2,1,1)= 0.0 # force My
#sSolmF(:,1,2)= 0.0

#sSolmF(:,1,3)= 0.0
#sSolmF(:,1,4)= 0.0
#sSolmF(:,1,5)= 0.0

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(2,1,SolmedeArv);

#tSiire(:,1,1)= 0.0
#tSiire(2,1,1)= 0.01*baasi0
#tSiire(:,1,2)= 0.0

#tSiire(:,1,3)= 0.0
#tSiire(:,1,4)= 0.0
#tSiire(:,1,5)= 0.0

##
otherwise
disp(' No load variant cases ')
endswitch
##

```

Program excerpt 3.11 (`spESTbeamLaheWFI.m`)

```

=====
#      Nodal coordinates
#=====
krdn=[#  x
      0.0 ; # node 1
      8.0 ; # node 2
      16.0 ; # node 3
      22.0 ; # node 4
      24.0]; # node 5
#=====
#
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No  w  fi
#=====
tsolm=[1  1 1 ; # node 1
       2  1 0 ; # node 2
       3  1 0 ; # node 3
       4  1 0 ]; # node 4
#=====

```

Program excerpt 3.12 (`spESTbeamLaheWFI.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#          n2 - end of the element
#          n1 - beginning of the element
#          N, Q, M - hinges at the end of the element
#          N, Q, M - hinges at the beginning of the element
#
EI GA  2  1  0 0  0 0;  % element 1
EI GA  3  2  0 0  0 0;  % element 2
EI GA  4  3  0 0  0 0;  % element 3
EI GA  5  4  0 0  0 0]; % element 4
# 1 - hinge 'true' (shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (3.14), carried out by the function `LaheBeamDFI(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the beam element ends as shown in Fig. 3.22. The unscaled initial parameter vectors of the elements are shown in excerpt 3.10 from the computing diary.*

Computing diary excerpt 3.10 (load case 1 `spESTbeamLaheWFI.m`)

```
-- Scaling multiplier for displacements = 1/baasi0 ---
```

```

=====
Unscaled initial parameter vector
Element      w      fi      Q      M
No
-----
1  +0.000e+00  +0.000e+00  -47.450  62.533
2  +0.000e+00  -2.933e-04  -49.650  66.933

```

3	+0.000e+00	+1.173e-03	-40.956	53.733
4	+0.000e+00	+3.133e-04	-24.000	24.000

The support reactions of the beam in global coordinates are shown in excerpt 3.11 from the computing diary.

Computing diary excerpt 3.11 (load case 1 spESTbeamLaheWFI.m)

Support reactions begin from row X: 33

```
=====
No          X          Node    Cz <=> 1
                          Cy <=> 2
-----
33  -4.745000e+01    1         1
34  +6.253333e+01    1         2
35  -9.820000e+01    2         1
36  -8.730556e+01    3         1
37  -5.504444e+01    4         1
-----
```

3. Output

Output of load case 1: the element displacements and forces determined by the transfer matrix are shown in excerpt 3.12 from the computing diary.

The shear force Q and the bending moment diagrams of the beam are shown in Fig. 3.23.

Testing a static equilibrium for the beam

Consider a static equilibrium of the beam shown in Fig. 3.23. Let us project the forces onto the Z-axis:

$$\Sigma Z = 0; \quad 12 \cdot 24 - 47.450 - 98.200 - 87.30556 - 55.04444 = 0 \quad (3.15)$$

The sum of the moments and the moments of the forces acting about point 1:

$$\Sigma M_1 = 0; \quad 62.53333 - 12 \cdot 24 + 98.20 \cdot 8 + 87.30556 \cdot 16 + 55.04444 \cdot 22 = -3.0000 \cdot 10^{-5} \text{ kN}\cdot\text{m} \approx 0 \quad (3.16)$$

The calculations with equations (3.15) and (3.16) have verified the static equilibrium of the beam.

Computing diary excerpt 3.12 (load case 1 spESTbeamLaheWFI.m)

```
=====
Element displacements and forces determined by transfer matrix
=====
Displacements and forces of element no 1 of length 8.000 m
The element is divided into 5 parts
displacement w - 0.000e+00  2.546e-03  5.673e-03  5.560e-03  2.321e-03  1.388e-17
rotation      fi - 0.000e+00 -2.375e-03 -1.135e-03  1.264e-03  2.364e-03 -2.933e-04
shear force   Q -  47.450    28.250    9.050    -10.150   -29.350   -48.550
moment force  M - -62.533    -1.973    27.867    26.987    -4.613   -66.933
```

 Displacements and forces of element no 2 of length 8.000 m

The element is divided into 5 parts

displacement w	-	0.000e+00	3.222e-03	7.137e-03	7.475e-03	3.898e-03	-1.388e-17
rotation fi	-	-2.933e-04	-2.880e-03	-1.569e-03	1.182e-03	2.915e-03	1.173e-03
shear force Q	-	49.650	30.450	11.250	-7.950	-27.150	-46.350
moment force M	-	-66.933	-2.853	30.507	33.147	5.067	-53.733

Displacements and forces of element no 3 of length 6.000 m

The element is divided into 5 parts

displacement w	-	0.000e+00	-1.152e-05	1.033e-03	1.461e-03	8.448e-04	6.939e-18
rotation fi	-	1.173e-03	-7.491e-04	-7.595e-04	1.053e-04	8.085e-04	3.133e-04
shear force Q	-	40.956	26.556	12.156	-2.244	-16.644	-31.044
moment force M	-	-53.733	-13.227	10.000	15.947	4.613	-24.000

Displacements and forces of element no 4 of length 2.000 m

The element is divided into 5 parts

displacement w	-	0.000e+00	-4.149e-05	4.117e-05	1.942e-04	3.793e-04	5.733e-04
rotation fi	-	3.133e-04	-7.707e-05	-3.139e-04	-4.355e-04	-4.803e-04	-4.867e-04
shear force Q	-	24.000	19.200	14.400	9.600	4.800	0.000
moment force M	-	-24.000	-15.360	-8.640	-3.840	-0.960	0.000

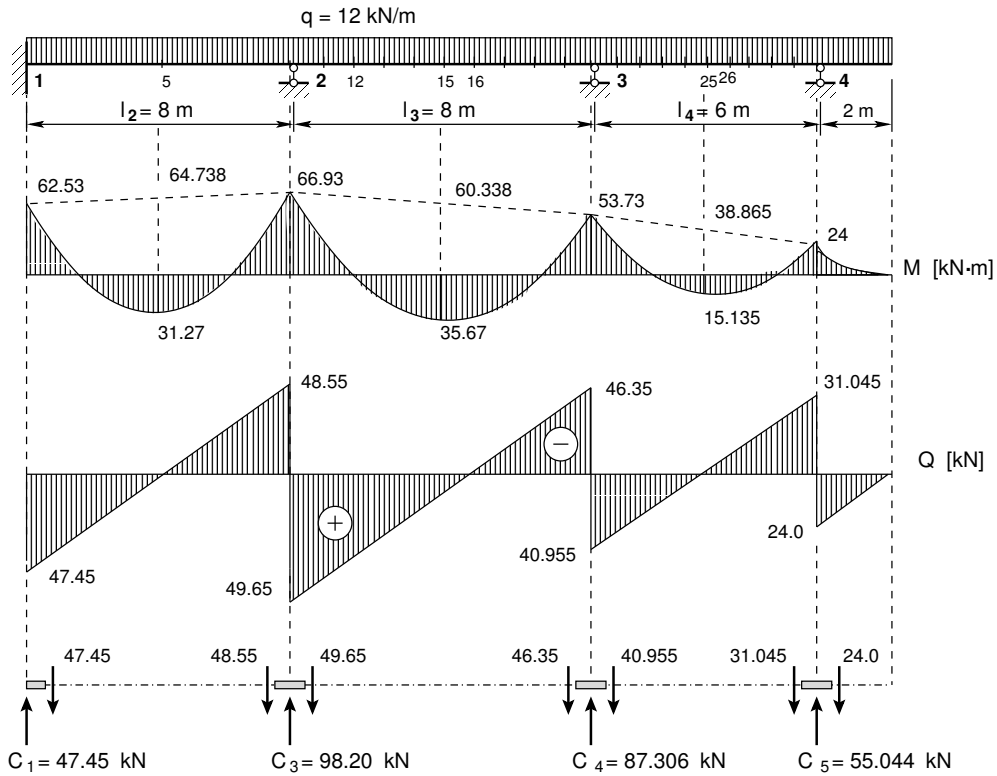


Figure 3.23. Internal forces diagrams of the continuous beam (1st loading)

```

Forces of element 1 at x = 4.000      shear force Q - 1.65000
displacement w - 6.10667e-03          moment force M - 35.66667
rotation fi - 7.33333e-05
shear force Q - -0.55000
moment force M - 31.26667
-----
Forces of element 2 at x = 1.599      shear force Q - -7.95000
displacement w - 3.22218e-03          moment force M - 33.14667
rotation fi - -2.88000e-03
shear force Q - 30.45001
moment force M - -2.85336
-----
Forces of element 2 at x = 4.000      shear force Q - -2.24444
displacement w - 7.86667e-03          moment force M - 15.94667
rotation fi - -2.20000e-04
-----

```

Output of load case 2: the initial parameter vector and support reactions.

Computing diary excerpt 3.13 (load case 2 `spESTbeamLaheWFI.m`)

-- Scaling multiplier for displacements = 1/baasi0 ---

```

=====
Unscaled initial parameter vector
Element      w          fi          Q          M
No
-----
1  +0.000e+00  +0.000e+00    10.752    -28.672
2  +0.000e+00  -5.734e-03   -65.536     57.344
3  +0.000e+00  +4.506e-03    -7.509     45.056
4  +0.000e+00  -2.253e-03     0.000     0.000
-----

```

The support reactions of the beam in global coordinates are shown in excerpt 3.14 from the computing diary.

Computing diary excerpt 3.14 (load case 2 `spESTbeamLaheWFI.m`)

Support reactions begin from row X: 33

```

=====
No          X          Node      Cz <=> 1
           Cy <=> 2
-----
33  +1.075200e+01    1          1
34  -2.867200e+01    1          2
35  -7.628800e+01    2          1
36  -4.197333e+01    3          1
37  +7.509333e+00    4          1
-----

```

The element displacements and forces determined by the transfer matrix are shown in excerpt 3.15 from the computing diary.

The shear force Q and the bending moment diagrams of the beam are shown in Fig. 3.24.

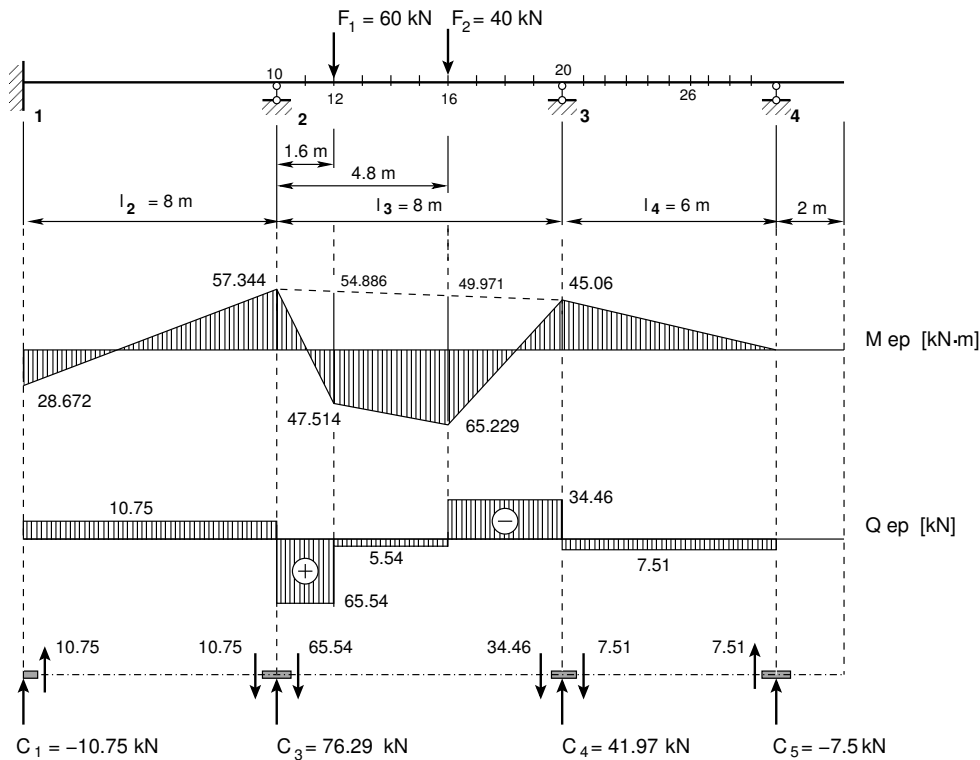


Figure 3.24. Internal forces diagrams of the continuous beam (2nd loading)

Testing a static equilibrium for the beam

Consider next a static equilibrium of the beam shown in Fig. 3.24. Let us project the forces onto the Z-axis:

$$\Sigma Z = 0; \quad \Sigma Z = 60.0 + 40.0 + 10.75 - 76.29 - 41.97 + 7.51 = 0 \quad (3.17)$$

The sum of the moments and the moments of the forces acting about point 1:

$$\Sigma M_1 = 0; \quad -28.672 - 60 \cdot 9.6 - 40 \cdot 12.8 + 76.288 \cdot 8 + 41.973 \cdot 16 - 7.5093 \cdot 22 = 2.0000 \cdot 10^{-4} \text{ kN}\cdot\text{m} \approx 0 \quad (3.18)$$

The calculations with equations (3.17) and (3.18) have verified the static equilibrium of the beam.

Computing diary excerpt 3.15 (load case 2 spESTbeamLaheWFL.m)

```

=====
Element displacements and forces determined by transfer matrix
=====
Displacements and forces of element no 1 of length 8.000 m
The element is divided into 5 parts
displacement w - 0.000e+00 -1.468e-03 -4.404e-03 -6.606e-03 -5.872e-03 -6.939e-18
rotation fi - 0.000e+00 1.606e-03 1.835e-03 6.881e-04 -1.835e-03 -5.734e-03
shear force Q - -10.752 -10.752 -10.752 -10.752 -10.752 -10.752
moment force M - 28.672 11.469 -5.734 -22.938 -40.141 -57.344
=====
    
```


Displacements and forces of element no 2 of length 8.000 m

The element is divided into 5 parts

```

displacement w - 0.000e+00 1.061e-02 1.718e-02 1.654e-02 8.916e-03 2.776e-17
rotation fi - -5.734e-03 -6.128e-03 -1.972e-03 2.892e-03 5.904e-03 4.506e-03
shear force Q - 65.536 5.536 5.536 -34.464 -34.464 -34.464
moment force M - -57.344 47.514 56.371 65.229 10.086 -45.056
    
```

Displacements and forces of element no 3 of length 6.000 m

The element is divided into 5 parts

```

displacement w - 0.000e+00 -3.893e-03 -5.190e-03 -4.542e-03 -2.595e-03 -6.939e-18
rotation fi - 4.506e-03 2.073e-03 1.802e-04 -1.171e-03 -1.982e-03 -2.253e-03
shear force Q - 7.509 7.509 7.509 7.509 7.509 7.509
moment force M - -45.056 -36.045 -27.034 -18.022 -9.011 0.000
    
```

Displacements and forces of element no 4 of length 2.000 m

The element is divided into 5 parts

```

displacement w - 0.000e+00 9.011e-04 1.802e-03 2.703e-03 3.604e-03 4.506e-03
rotation fi - -2.253e-03 -2.253e-03 -2.253e-03 -2.253e-03 -2.253e-03 -2.253e-03
shear force Q - 0.000 0.000 0.000 0.000 0.000 0.000
moment force M - 0.000 0.000 0.000 0.000 0.000 0.000
    
```

Forces of element 1 at x = 4.000

```

displacement w - -5.73440e-03
rotation fi - 1.43360e-03
shear force Q - -10.75200
moment force M - -14.33600
    
```

Forces of element 2 at x = 1.599

```

displacement w - 1.06081e-02
rotation fi - -6.12762e-03
shear force Q - 65.53600
moment force M - 47.51353
    
```

Forces of element 2 at x = 4.000

```

displacement w - 1.78347e-02
rotation fi - 3.71200e-04
shear force Q - 5.53600
moment force M - 60.80000
    
```

Forces of element 3 at x = 3.600

```

displacement w - -4.54164e-03
rotation fi - -1.17146e-03
shear force Q - 7.50933
moment force M - -18.02240
    
```

Forces of element 2 at x = 4.800

```

displacement w - 1.65413e-02
rotation fi - 2.89178e-03
shear force Q - -34.46400
moment force M - 65.22880
    
```

Output of load case 3: the initial parameter vector and support reactions.

Computing diary excerpt 3.16 (load case 3 `spESTbeamLaheWFL.m`)

-- Scaling multiplier for displacements = 1/baasi0 ---

=====

Unscaled initial parameter vector

Element No	w	fi	Q	M
1	+0.000e+00	+0.000e+00	-2.016	5.376

2	+0.000e+00	+1.075e-03	6.048	-10.752
3	+0.000e+00	-4.301e-03	-38.272	37.632
4	+0.000e+00	+7.334e-03	0.000	0.000

The support reactions of the beam in global coordinates are shown in excerpt 3.17 from the computing diary.

Computing diary excerpt 3.17 (load case 3 spESTbeamLaheWFI.m)

Support reactions begin from row X: 33

No	X	Node	Cz <=> 1	Cy <=> 2
33	-2.016000e+00	1	1	
34	+5.376000e+00	1		2
35	+8.064000e+00	2	1	
36	-4.432000e+01	3	1	
37	-4.172800e+01	4	1	

The element displacements and forces determined by the transfer matrix are shown in excerpt 3.18 from the computing diary.

The shear force Q and the bending moment diagrams of the beam are shown in Fig. 3.25.

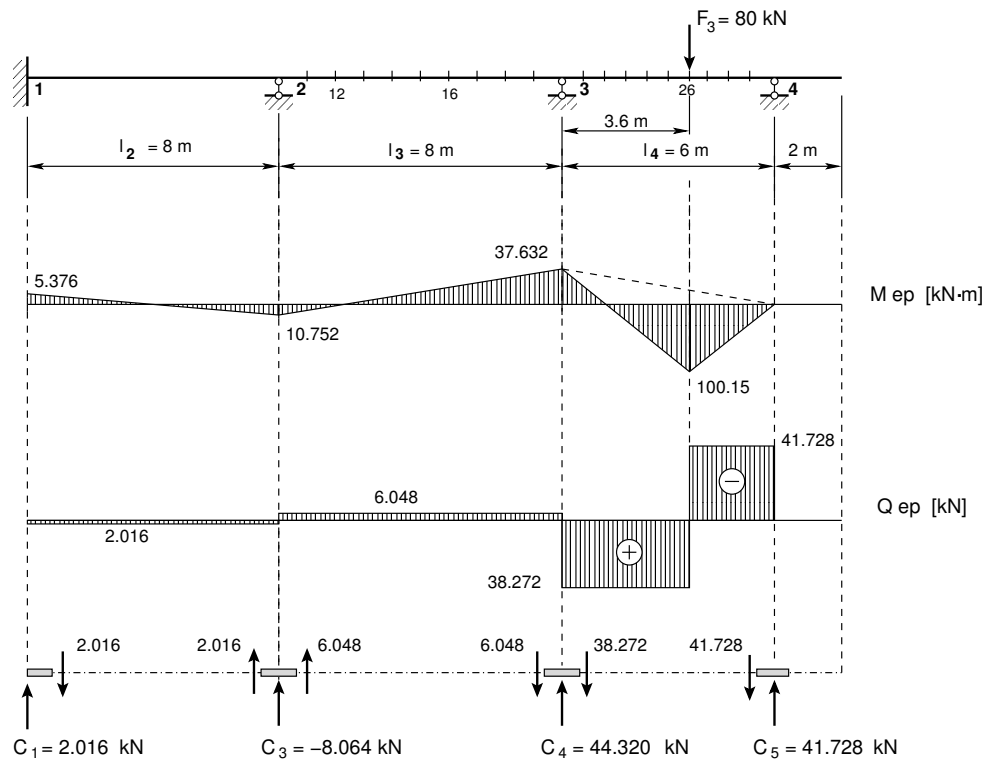


Figure 3.25. Internal forces diagrams of the continuous beam (3rd loading)

Testing a static equilibrium for the beam

Consider now a static equilibrium of the beam shown in Fig. 3.25. Let us project the forces onto the Z-axis:

$$\Sigma Z = 0; \quad \Sigma Z = 80 - 2.016 + 8.064 - 44.320 - 41.728 = 0 \quad (3.19)$$

The sum of the moments and the moments of the forces acting about point 1:

$$\begin{aligned} \Sigma M_1 = 0; \quad & 5.376 - 80 \cdot 19.6 - 8.064 \cdot 8 + 44.320 \cdot 16 \\ & + 41.728 \cdot 22 = 1.1369 \cdot 10^{-13} \text{ kN}\cdot\text{m} \approx 0 \end{aligned} \quad (3.20)$$

The calculations with equations (3.19) and (3.20) have verified the static equilibrium of the beam.

Computing diary excerpt 3.18 (load case 3 spESTbeamLaheWFL.m)

```

=====
Element displacements and forces determined by transfer matrix
=====
Displacements and forces of element no 1 of length 8.000 m
The element is divided into 5 parts
displacement w - 0.000e+00  2.753e-04  8.258e-04  1.239e-03  1.101e-03  -1.735e-18
rotation      fi - 0.000e+00 -3.011e-04 -3.441e-04 -1.290e-04  3.441e-04  1.075e-03
shear force   Q -   2.016    2.016    2.016    2.016    2.016    2.016
moment force  M -  -5.376   -2.150    1.075    4.301    7.526   10.752
-----
Displacements and forces of element no 2 of length 8.000 m
The element is divided into 5 parts
displacement w - 0.000e+00 -2.202e-03 -4.542e-03 -5.780e-03 -4.679e-03  0.000e+00
rotation      fi - 1.075e-03  1.548e-03  1.247e-03  1.720e-04 -1.677e-03 -4.301e-03
shear force   Q -  -6.048   -6.048   -6.048   -6.048   -6.048   -6.048
moment force  M -  10.752    1.075   -8.602  -18.278  -27.955  -37.632
-----
Displacements and forces of element no 3 of length 6.000 m
The element is divided into 5 parts
displacement w - 0.000e+00  5.965e-03  1.133e-02  1.280e-02  8.200e-03 -3.469e-18
rotation      fi - -4.301e-03 -5.181e-03 -3.305e-03  1.326e-03  5.832e-03  7.334e-03
shear force   Q -  38.272   38.272   38.272   38.272  -41.728  -41.728
moment force  M - -37.632    8.294   54.221  100.147  50.074   0.000
-----
Displacements and forces of element no 4 of length 2.000 m
The element is divided into 5 parts
displacement w - 0.000e+00 -2.934e-03 -5.868e-03 -8.801e-03 -1.174e-02 -1.467e-02
rotation      fi - 7.334e-03  7.334e-03  7.334e-03  7.334e-03  7.334e-03  7.334e-03
shear force   Q -   0.000    0.000    0.000    0.000    0.000    0.000
moment force  M -   0.000    0.000    0.000    0.000    0.000    0.000
-----

Forces of element 1 at x = 4.000
displacement w - 1.07520e-03
rotation      fi - -2.68800e-04
shear force   Q -  2.01600
moment force  M -  2.68800
-----

```

Forces of element 2 at x = 1.599
 displacement w - -2.20201e-03
 rotation fi - 1.54829e-03
 shear force Q - -6.04800
 moment force M - 1.07521

Forces of element 2 at x = 4.800
 displacement w - -5.78028e-03
 rotation fi - 1.72032e-04
 shear force Q - -6.04800
 moment force M - -18.27840

Forces of element 2 at x = 4.000
 displacement w - -5.37600e-03
 rotation fi - 8.06400e-04
 shear force Q - -6.04800
 moment force M - -13.44000

Forces of element 3 at x = 3.600
 displacement w - 1.27955e-02
 rotation fi - 1.32557e-03
 shear force Q - -41.72800
 moment force M - 100.14720

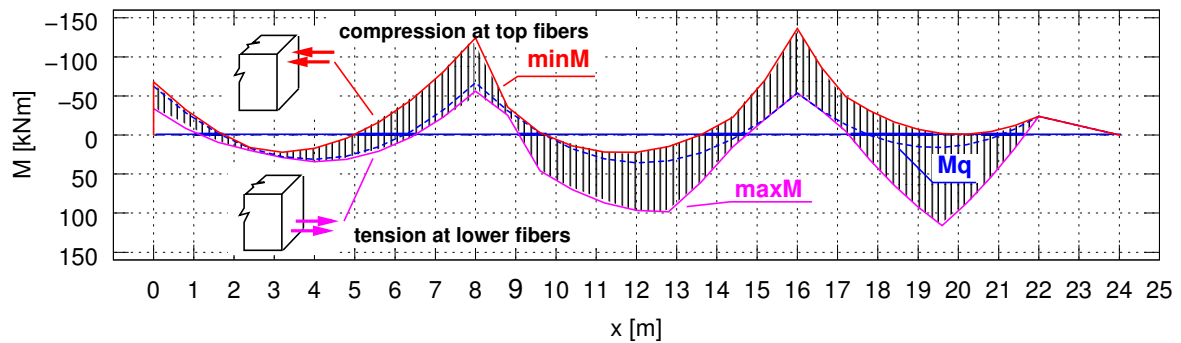


Figure 3.26. The continuous beam minimum-maximum bending moments²

=====
 Minimum-maximum bending moments
 =====

No	x	Mq	Mmax	Min
1	0.00	-62.53	-33.86	-67.91
2	0.80	-28.41	-8.34	-32.18
3	1.60	-1.97	9.50	-4.12
4	2.40	16.79	19.65	16.25
5	3.20	27.87	28.94	22.13
6	4.00	31.27	33.95	16.93
7	4.80	26.99	31.29	4.05
8	5.60	15.03	20.94	-16.51
9	6.40	-4.61	2.91	-44.75
10	7.20	-31.93	-22.79	-80.68
11	8.00	-66.93	-56.18	-124.28
12	8.80	-31.05	-25.14	-35.97
13	9.60	-2.85	45.74	-2.85
14	10.40	17.67	69.61	13.90
15	11.20	30.51	86.88	21.91
16	12.00	35.67	96.47	22.23
17	12.80	33.15	98.38	14.87
18	13.60	22.95	60.60	-0.17
19	14.40	5.07	15.15	-22.89
20	15.20	-20.49	-20.49	-70.77
21	16.00	-53.73	-53.73	-136.42
22	16.60	-31.32	-31.32	-86.54
23	17.20	-13.23	-4.93	-49.27
24	17.80	0.55	31.80	-30.99
25	18.40	10.00	64.22	-17.03
26	19.00	15.13	92.32	-7.39
27	19.60	15.95	116.09	-2.08
28	20.20	12.44	87.55	-1.08
29	20.80	4.61	54.69	-4.40
30	21.40	-7.53	17.50	-12.04
31	22.00	-24.00	-24.00	-24.00
32	24.00	0.00	0.00	0.00

²http://digi.lib.ttu.ee/opik_eme/Ehitusmehaanika.pdf#page=278 [Lah12].

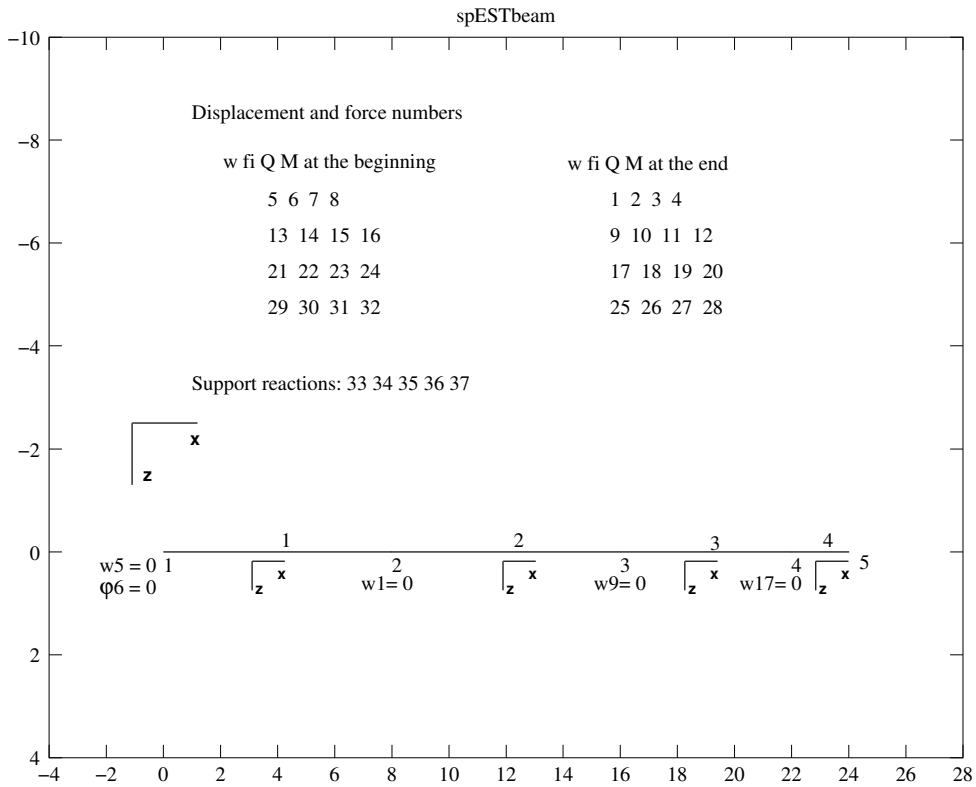


Figure 3.27. Elements of the continuous beam EST

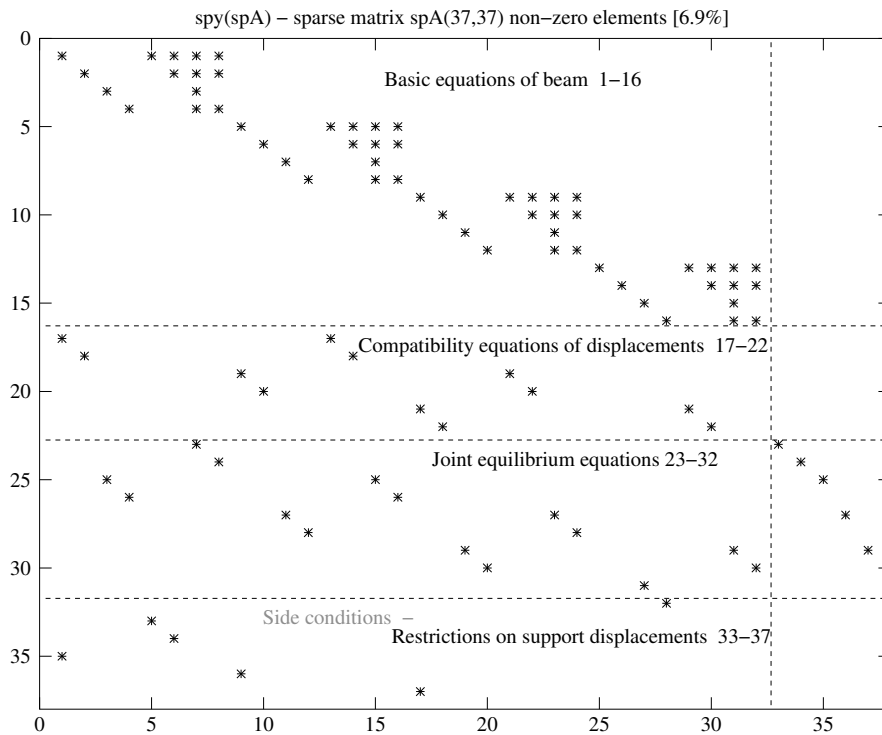


Figure 3.28. Sparsity pattern of matrix spA of the continuous beam EST

3.5 Illustrative truss problem

Example 3.5. Problem Statement. Given the truss of Fig. 3.29, find the reactions, displacements and internal forces. The height of the truss $H = 2.25$ m and the span length $L = 12$ m. The panel length $d = 3$ m. The dead load $F = 12$ kN reacts on the upper side nodes.

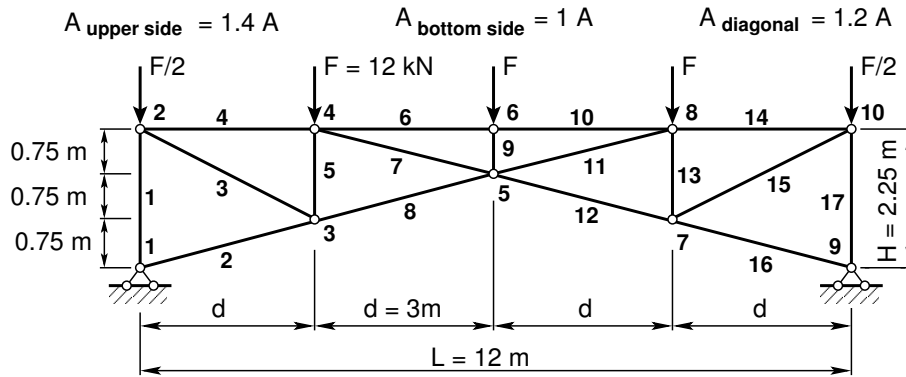


Figure 3.29. The truss EST

Let us assume that the cross-sectional area of the upper side members $A_y = 1.4 \cdot A \text{ cm}^2$ where $A = 9.58 \cdot 10^{-4} \text{ m}^2$ (9.58 cm^2), the cross-sectional area of the bottom side members $A_a = 1.0 \cdot A \text{ cm}^2$ and that of the diagonal members $A_a = 1.2 \cdot A \text{ cm}^2$. The modulus of elasticity $E = 2.1 \cdot 10^{11} \text{ Pa}$.

Problem Statement. To solve the problem, we use the EST method. The solving procedure includes the following.

1. Data input: the number of truss nodes, elements, support reactions; element properties, node forces in global coordinates, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (3.21)$$

2. Assembling and solving the boundary problem equations (2.1) and (3.21), carried out by the function `LaheTrussDFI.m`. To implement this aim, the program

- (a) inserts the basic equations of truss into the equation system,
- (b) adds the compatibility equations of the displacements at nodes to the equation system,
- (c) adds the joint equilibrium equations,
- (d) adds the restrictions on support displacements,
- (e) solves the compiled system of sparse equations,
- (f) produces an output: initial parameter vectors for element displacements and forces; support reactions.

3. Output: the displacements and forces of truss members.

1. **Input data** for the GNU Octave program `spESTtrussLaheWFI.m` are shown in excerpts from the program: element and nodal loads – excerpt 3.13; nodal coordinates – excerpt 3.14; element properties, topology and hinges – excerpt 3.15.

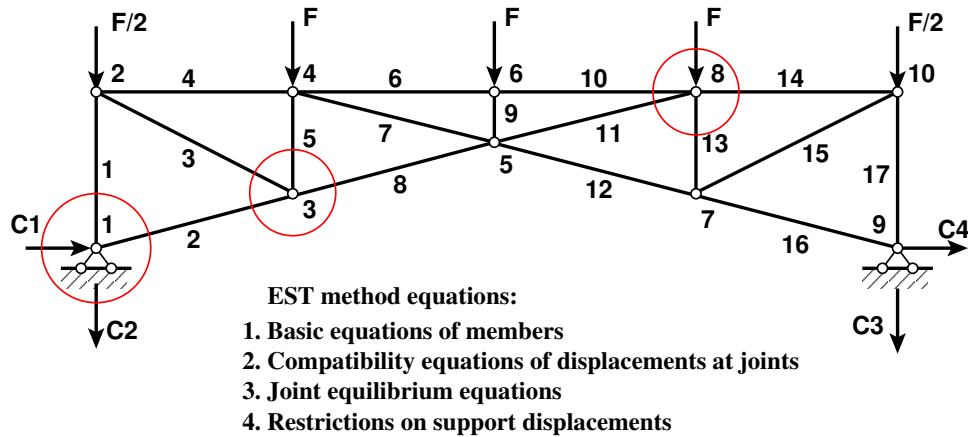


Figure 3.30. A free-body diagram of the truss with joint numbers

Program excerpt 3.13 (spESTtrussLaheWFI.m)

```

Number_of_truss_nodes=10
Number_of_elements=17
Number_of_support_reactions=4
d=3.0; # panel length
Npanel=4; # panel number
L=12.0; # truss span
H=2.25; # truss height
spNNK=8*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Lp=12.0; # graphics axis
# --- Element properties ---

E=2.1E+11 # Pa                      EAy=Ay*E;
A=9.58E-04 # A=9.58 cm2             EAd=Ad*E;
EAa=Aa*E;

baasi0=EAy/d # scaling multiplier for displacements

#disp(' Node forces in global coordinates ')
# sSolmF(forces,1,nodes); forces=[Fx; Fz]
sSolmF = zeros(2,1,SolmedeArv);

sSolmF = zeros(2,1,SolmedeArv);      sSolmF(2,1,6)=Fz;    # s6F(2,1)=F;
#sSolmF(:,1,1)=0.0                   #sSolmF(:,1,7)=0.0
sSolmF(2,1,2)=Fz/2;    # s2F(2,1)=F/2;    sSolmF(2,1,8)= Fz;    # s8F(2,1)=F;
#sSolmF(:,1,3)=0.0                   #sSolmF(:,1,9)=0.0
sSolmF(2,1,4)=Fz;    # s4F(2,1)=F;        sSolmF(2,1,10)=Fz/2; # s10F(2,1)=F/2;
#sSolmF(:,1,5)=0.0

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(2,1,SolmedeArv);

```

```
#tSiire(1,1,1)= 0.0                #tSiire(1,1,9)= 0.0
#tSiire(2,1,1)= 0.01*baasi0        #tSiire(2,1,9)= 0.0
```

Program excerpt 3.14 (`spESTtrussLaheWFI.m`)

```
#####
#      Nodal coordinates
#####

krdn=[#  x      z
      0.0    0.0 ; % node 1          6.0    -2.25 ; % node 6
      0.0   -2.25 ; % node 2          9.0    -0.75 ; % node 7
      3.0   -0.75 ; % node 3          9.0    -2.25 ; % node 8
      3.0   -2.25 ; % node 4         12.0     0.0 ; % node 9
      6.0   -1.5  ; % node 5         12.0   -2.25] ; % node 10

#####
#
#####
# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#####
tsolm=[1    1 1 0; % node 1
      9    1 1 0]; % node 9
#####
```

Program excerpt 3.15 (`spESTtrussLaheWFI.m`)

```
# ----- Element properties, topology -----
elasts=[# Element properties
#      n2 - end of the element
#      n1 - beginning of the element
#      A  E;

      2  1  Ad  E; % element 1          8  6  Ay  E; % element 10
      3  1  Aa  E; % element 2          8  5  Ad  E; % element 11
      3  2  Ad  E; % element 3          7  5  Aa  E; % element 12
      4  2  Ay  E; % element 4          8  7  Ad  E; % element 13
      4  3  Ad  E; % element 5         10  8  Ay  E; % element 14
      6  4  Ay  E; % element 6         10  7  Ad  E; % element 15
      5  4  Ad  E; % element 7          9  7  Aa  E; % element 16
      5  3  Aa  E; % element 8         10  9  Ad  E]; % element 17
      6  5  Ad  E; % element 9
```

2. *Assembling and solving the boundary problem equations (3.21), carried out by the function `LaheTrussDFI(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the truss element ends as shown in Fig. 3.31. The unscaled initial parameter vectors of the elements are shown in excerpt 3.19 from the computing diary.*

Computing diary excerpt 3.19 (load case 1 `spESTtrussLaheWFI.m`)

```
-- Scaling multiplier for displacements = 1/baasi0 ---
```



```
=====
Unscaled initial parameter vector (Sign Convention 2)
Element      u      w      fi      N
No
-----
```

Element No	u	w	fi	N
1	0.000e+00	-0.000e+00	-2.056e-04	15142.038
2	0.000e+00	0.000e+00	-5.898e-04	36522.315
3	4.768e-04	-8.061e-05	-5.458e-04	-20442.218
4	4.625e-04	1.411e-04	-6.166e-04	18284.075
5	-1.906e-03	-1.022e-04	-2.467e-04	13713.056
6	2.677e-04	1.991e-03	-4.938e-04	25136.301
7	7.426e-04	1.866e-03	-4.740e-04	-7063.113
8	-5.614e-04	1.824e-03	-4.878e-04	17675.521
9	-3.435e-03	-2.132e-21	-1.083e-19	12000.000
10	7.911e-20	3.472e-03	4.938e-04	25136.301
11	-8.331e-04	3.332e-03	4.740e-04	-7063.113
12	8.331e-04	3.332e-03	4.878e-04	17675.521
13	-1.906e-03	1.022e-04	2.467e-04	13713.056
14	-2.677e-04	1.991e-03	6.166e-04	18284.075
15	-7.608e-04	1.750e-03	5.458e-04	-20442.218
16	5.614e-04	1.824e-03	5.898e-04	36522.315
17	-0.000e+00	0.000e+00	2.056e-04	15142.038

```
-----
```

The support reactions of the truss in global coordinates are shown in excerpt 3.20 from the computing diary.

Computing diary excerpt 3.20 (load case 1 `spESTtrussLaheWFI.m`)
 Support reactions begin from row X: 33

```
=====
```

No	X	Node	Cx <=> 1	Cz <=> 2
137	+3.543185e+04	1	1	
138	-2.400000e+04	1		2
139	-3.543185e+04	9	1	
140	-2.400000e+04	9		2

```
-----
```

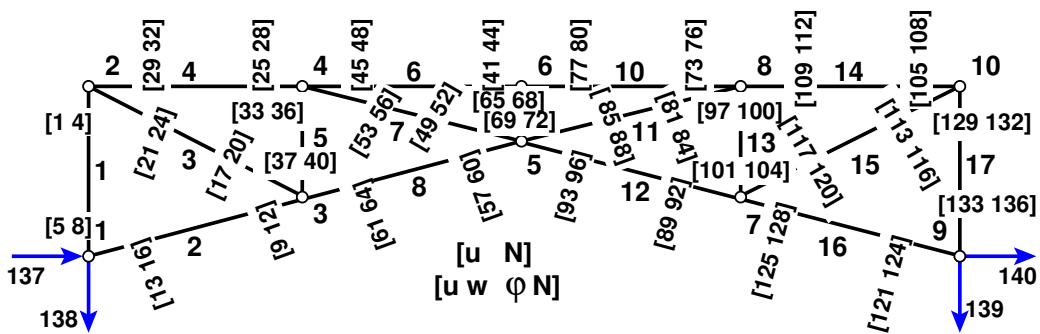


Figure 3.31. Numeration of displacements and forces of the truss EST

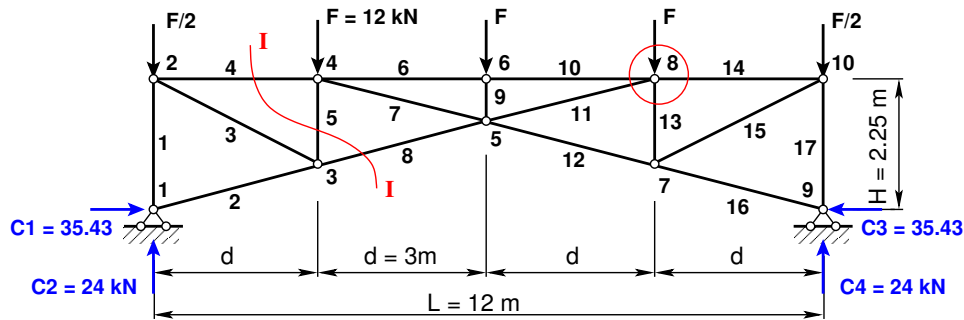


Figure 3.32. Support reactions of the truss EST

3. Output

Results of computation. The values of displacements and forces of the truss members are presented in Table 3.1.

The verifiable results have been obtained with the finite element method³ (FEM). The displacements determined by the EST method are u^* , w^* (in local coordinates), these determined by the FEM are u , w (in global coordinates). The displacements u^* , w^* of members 1, 5, 9, 13, and 17 compared with the displacements u , w of nodes 1, 2, 3, ..., 10 are shown in Table 3.1.

The forces and displacements determined by the EST method are of the same value as these determined by the FEM.

Testing a static equilibrium for the truss

Consider next a static equilibrium of the truss shown in Fig. 3.32. Let us project the forces onto the X-axis,

$$\Sigma X = 0; \quad 35.43 - 35.43 = 0 \quad (3.22)$$

and onto the Z-axis,

$$\Sigma Z = 0; \quad 4 \cdot 12.0 - 2 \cdot 24.0 = 0 \quad (3.23)$$

Now we consider the left part of the truss (members 4, 5, 8 are cut through with the cut I-I) shown in Fig. 3.32. Here, the sum of the moments of the forces acting about point 8 is

$$\Sigma M_1 = 0; \quad (-24.0 + 6.0) 9.0 + 13.713 \cdot 6.0 + 35.432 \cdot 2.25 = 0 \quad (3.24)$$

where the internal force $N_5 = -13.713$ kN.

The calculations with equations (3.22), (3.23), and (3.24) have verified the static equilibrium of the truss.

³http://digi.lib.ttu.ee/opik_eme/Ehitusmehaanika.pdf#page=394 Table 14.1.

Table 3.1. Comparison of the internal forces of the truss

<i>Member No</i>	EST method (Sign Convention 1)			FEM (global coordinates)		
	u^* [m]	w^* [m]	N [N]	u [m]	w [m]	N [kN]
1 <i>begin.</i>	0.000e+00	0.000e+00	-15142.038	0.000e+00	0.000e+00	
1 <i>end</i>	-1.411e-04	4.625e-04	-15142.038	4.625e-02	1.411e-02	-15.1420
2 <i>begin.</i>	-0.000e+00	-0.000e+00	-36522.315			
2 <i>end</i>	-5.614e-04	1.824e-03	-36522.315			-36.5223
3 <i>begin.</i>	4.768e-04	-8.061e-05	20442.218			
3 <i>end</i>	7.608e-04	1.750e-03	20442.218			20.4422
4 <i>begin.</i>	4.625e-04	1.411e-04	-18284.075			
4 <i>end</i>	2.677e-04	1.991e-03	-18284.075			-18.2841
5 <i>begin.</i>	-1.906e-03	-1.022e-04	-13713.056	-1.022e-02	1.906e-01	
5 <i>end</i>	-1.991e-03	2.677e-04	-13713.056	2.677e-02	1.991e-01	-13.7131
6 <i>begin.</i>	2.677e-04	1.991e-03	-25136.301			
6 <i>end</i>	3.056e-20	3.472e-03	-25136.301			-25.1363
7 <i>begin.</i>	7.426e-04	1.866e-03	7063.113			
7 <i>end</i>	8.331e-04	3.332e-03	7063.113			7.0631
8 <i>begin.</i>	-5.614e-04	1.824e-03	-17675.521			
8 <i>end</i>	-8.331e-04	3.332e-03	-17675.521			-17.6755
9 <i>begin.</i>	-3.435e-03	1.540e-21	-12000.000	-1.133e-16	3.435e-01	
9 <i>end</i>	-3.472e-03	3.056e-20	-12000.000	-1.196e-16	3.472e-01	-12.0000
10 <i>begin.</i>	3.056e-20	3.472e-03	-25136.301			
10 <i>end</i>	-2.677e-04	1.991e-03	-25136.301			-25.1363
11 <i>begin.</i>	-8.331e-04	3.332e-03	7063.113			
11 <i>end</i>	-7.426e-04	1.866e-03	7063.113			7.0631
12 <i>begin.</i>	8.331e-04	3.332e-03	-17675.521			
12 <i>end</i>	5.614e-04	1.824e-03	-17675.521			-17.6755
13 <i>begin.</i>	-1.906e-03	1.022e-04	-13713.056	1.022e-02	1.906e-01	
13 <i>end</i>	-1.991e-03	-2.677e-04	-13713.056	-2.677e-02	1.991e-01	-13.7131
14 <i>begin.</i>	-2.677e-04	1.991e-03	-18284.075			
14 <i>end</i>	-4.625e-04	1.411e-04	-18284.075			-18.2841
15 <i>begin.</i>	-7.608e-04	1.750e-03	20442.218			
15 <i>end</i>	-4.768e-04	-8.061e-05	20442.218			20.4422
16 <i>begin.</i>	5.614e-04	1.824e-03	-36522.315			
16 <i>end</i>	0.000e+00	0.000e+00	-36522.315			-36.5223
17 <i>begin.</i>	0.000e+00	0.000e+00	-15142.038	0.000e+00	0.000e+00	
17 <i>end</i>	-1.411e-04	-4.625e-04	-15142.038	-4.625e-02	1.411e-02	-15.1420

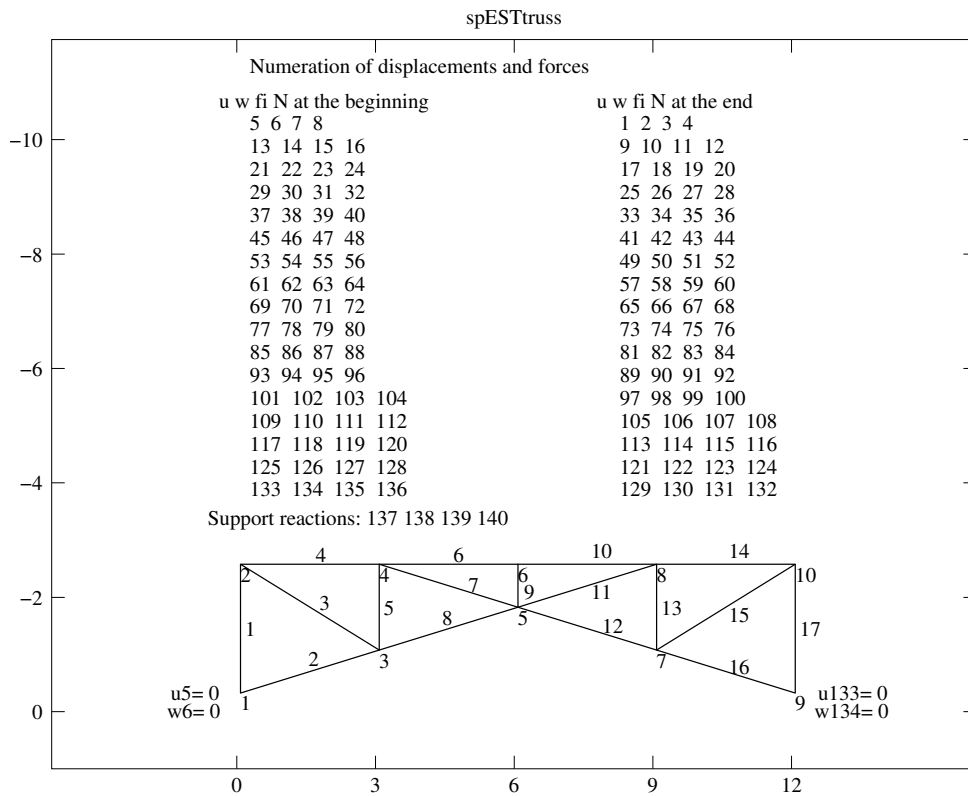


Figure 3.33. Elements of the truss EST

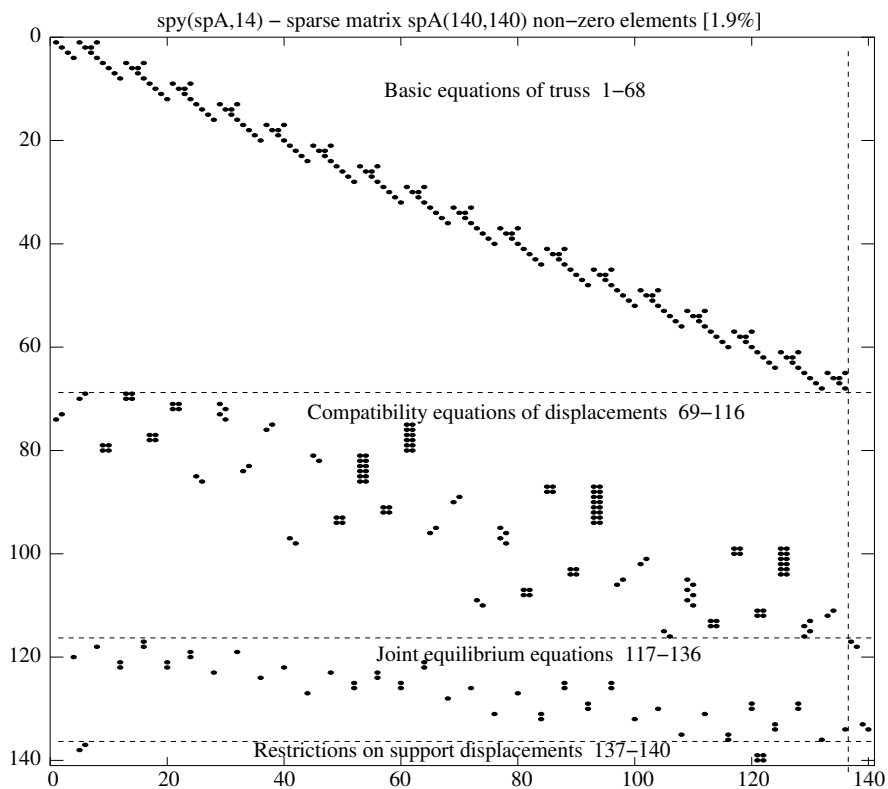


Figure 3.34. Sparsity pattern of matrix spA of the truss

4. Statically determinate problems

4.1 Illustrative frame problem

Example 4.1. Problem Statement. Figure 4.1 shows a frame with the dimensions as follows: height of the column $h = 4$ m, height of the frame 5 m, span length $l = 8$ m. The rafter 2-3 is loaded with a vertical uniform load $q = 2.0$ kN/m, the rafter 3-4 is loaded with a vertical concentrated load $F_1 = 3.0$ kN, and the joint 2 is subjected to a horizontal load $F_2 = 1.0$ kN.

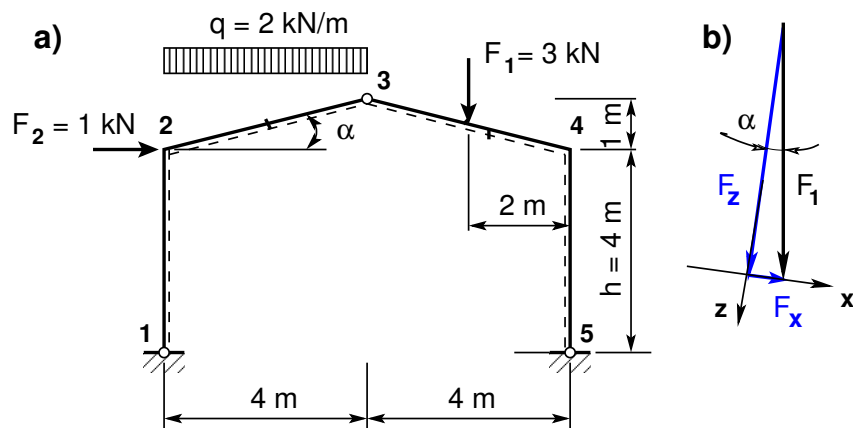


Figure 4.1. The three-hinged frame EST1

Consider the load transformation from the global system to the local one shown in Figs. 4.1 b and 4.2. The force F_1 is transformed to F_z , F_x :

$$\begin{aligned} aF_3 &= l_2/2, & \text{where } l_2 &= \sqrt{4^2 + 1.0^2} \\ F_z &= F_1 \cdot \cos \alpha, & \text{where } \cos \alpha &= 4.0/l_2 \\ F_x &= F_1 \cdot \sin \alpha, & \text{where } \sin \alpha &= 1.0/l_2 \end{aligned} \quad (4.1)$$

The vertical load q is transformed to q_x , q_z :

$$\begin{aligned} q_k &= q \cdot 4.0/l_2, & \text{where } l_2 &= \sqrt{4^2 + 1.0^2} \\ q_x &= -q_k \cdot \sin \alpha, & \text{where } \sin \alpha &= 1.0/l_2 \\ q_z &= q_k \cdot \cos \alpha, & \text{where } \cos \alpha &= 4.0/l_2 \end{aligned} \quad (4.2)$$

We wish to compute the reactions and internal forces, and draw the axial force, shear force and bending moment diagrams.

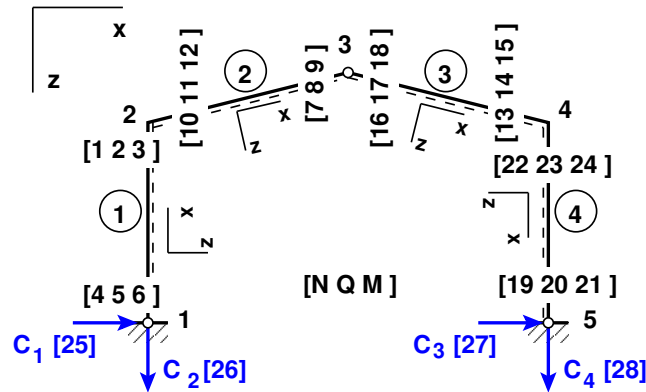


Figure 4.2. Numeration of displacements and forces of the frame EST1

Problem Solving. To solve the problem, we use the EST method. The solving procedure includes the following.

1. Data input: the number of frame nodes, elements, support reactions; element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (4.3)$$

2. Assembling and solving the boundary problem equations (4.3) (prepared and solved by the program):
 - (a) writing the basic equations of a frame in transfer matrix form,
 - (b) adding the joint equilibrium equations,
 - (c) adding the side conditions (hinges),
 - (d) solving the system of sparse equations,
 - (e) producing an output: initial parameter vectors for element forces; support reactions.
3. Output: element forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program `spESTframe3hingeLaheNQM.m` are given in excerpts from the program: element and nodal loads – excerpt 4.1; nodal coordinates – excerpt 4.2; element properties, topology and hinges – excerpt 4.3.

Program excerpt 4.1 (`spESTframe3hingeLaheNQM.m`)

```

Number_of_frame_nodes=5
Number_of_elements=4
Number_of_support_reactions=4
spNNK=6*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Forces are calculated on parts ("Nmitmeks") of the element
Nmitmeks=4
Lp=8.0; # graphics axes
# --- Element loads ---

```

```

F1=3.0;
L2x=sqrt(4.0^2+1.0^2); # length of
                        the element
Lpunkt=L2x/2;
F2=1.0;
q=2.0;
sinA2=1.0/L2x;
cosA2=4.0/L2x;
qk=2.0*4.0/L2x;
#esQkoormus qz2 qx2 0.0 L2x
qz2=qk*cosA2; # projection onto z-axis

qx2=-qk*sinA2; # projection onto x-axis
#
#esFjoud Fz3 Fx3 aF3
Fz3=F1*cosA2; # projection onto z-axis
Fx3=F1*sinA2; # projection onto x-axis
aF3=L2x/2;
# ---- load variants -----
load_variant=1
#load_variant=2
#load_variant=3

switch (koormusvariant)
case{1}
disp(' Load variant 1 ')

#Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,3)=[0.0 0.0 0.0 L2x];
esQkoormus(1,1:4,2)=[qz2 qx2 0.0 L2x]; esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 0.0]; esFjoud(1,1:3,4)=[0.0 0.0 0.0];
esFjoud(1,1:3,2)=[0.0 0.0 0.0]; esFjoud(1,1:3,5)=[0.0 0.0 0.0];
esFjoud(1,1:3,3)=[Fz3 Fx3 aF3];

#Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0 #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2; #sSolmF(:,1,5)= 0.0
#sSolmF(:,1,3)= 0.0

#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

case{2}
disp(' Load variant 2 ')
case{3}
disp(' Load variant 3 ')
#
otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 4.2 (spESTframe3hingeLaheNQM.m)

```

=====
#      Nodal coordinates
#=====
krdn=[#  x      z
      0.0    0.0;   # node 1
      0.0   -4.0;   # node 2
      4.0   -5.0;   # node 3
      8.0   -4.0;   # node 4
      8.0    0.0];  # node 5
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#=====
tsolm=[1    1 1 0;   % node 1
       5    1 1 0]; % node 5
#=====

```

Program excerpt 4.3 (spESTframe3hingeLaheNQM.m)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element
#      n2 - end of the element
#      n1 - beginning of the element
#      N, Q, M - hinges at the end of the element
#      N, Q, M - hinges at the beginning of the element
      2    1    0 0 0    0 0 1;   % element 1
      3    2    0 0 1    0 0 0;   % element 2
      4    3    0 0 0    0 0 1;   % element 3
      5    4    0 0 1    0 0 0];  % element 4
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (4.3), carried out by the function `LaheFrame3hingeNQM(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the forces at element ends of the frame as shown in Fig. 4.2.*

The initial parameter vectors of the elements are given in excerpt 4.1 from the computing diary.

Computing diary excerpt 4.1 (spESTframe3hingeLaheNQM.m)

```

=====
Initial parameter vector
Element No      N          Q          M
-----
      1          6.250        1.600        0.000
      2          4.038       -5.433        6.400
      3          2.947        1.067        0.000
      4          4.750       -2.600       10.400
-----

```

The support reactions of the frame in global coordinates are shown in excerpt 4.2 from the computing diary.

Computing diary excerpt 4.2 (spESTframe3hingeLaheNQM.m)

Support reactions begin from X row: 25

```
=====
No          X          Node    Cx <=> 1
                               Cz <=> 2
                               Cy <=> 3
-----
25  +1.600000e+00    1         1
26  -6.250000e+00    1         2
27  -2.600000e+00    5         1
28  -4.750000e+00    5         2
-----
```

3. Output: the forces determined by the transfer matrix are shown in excerpt 4.3 from the computing diary.

Computing diary excerpt 4.3 (spESTframe3hingeLaheNQM.m)

```
#=====
Element forces determined by transfer matrix
#=====
Forces of element no 1 of length 4.000 m
The element is divided into 4 parts
normal force N -   -6.250   -6.250   -6.250   -6.250   -6.250
shear force Q -   -1.600   -1.600   -1.600   -1.600   -1.600
moment force M -    0.000   -1.600   -3.200   -4.800   -6.400
-----
Forces of element no 2 of length 4.123 m
The element is divided into 4 parts
normal force N -   -4.038   -3.553   -3.068   -2.583   -2.098
shear force Q -    5.433    3.493    1.552   -0.388   -2.328
moment force M -   -6.400   -1.800    0.800    1.400   -0.000
-----
Forces of element no 3 of length 4.123 m
The element is divided into 4 parts
normal force N -   -2.947   -2.947   -3.674   -3.674   -3.674
shear force Q -   -1.067   -1.067   -3.978   -3.978   -3.978
moment force M -    0.000   -1.100   -2.200   -6.300  -10.400
-----
Forces of element no 4 of length 4.000 m
The element is divided into 4 parts
normal force N -   -4.750   -4.750   -4.750   -4.750   -4.750
shear force Q -    2.600    2.600    2.600    2.600    2.600
moment force M -  -10.400   -7.800   -5.200   -2.600   -0.000
-----
```

Testing a static equilibrium for the frame

Consider next a static equilibrium of the frame shown in Fig. 4.3. We project the forces onto the X, Z axes and determine the sum of the moments of the forces acting about point 1. The sums of the forces and moments are:

$$\begin{aligned}
 \Sigma X &= 0; & 1.6 + 1.0 - 2.6 &= 0 \\
 \Sigma Z &= 0; & 6.25 + 4.75 - 2 \cdot 4 - 3 &= 0 \\
 \Sigma M_1 &= 0; & 4.75 \cdot 8 - 1 \cdot 4 - 2 \cdot 4 \cdot 2 &= 0
 \end{aligned}
 \tag{4.4}$$

The calculations shown in equations (4.4) have verified the static equilibrium of the frame.

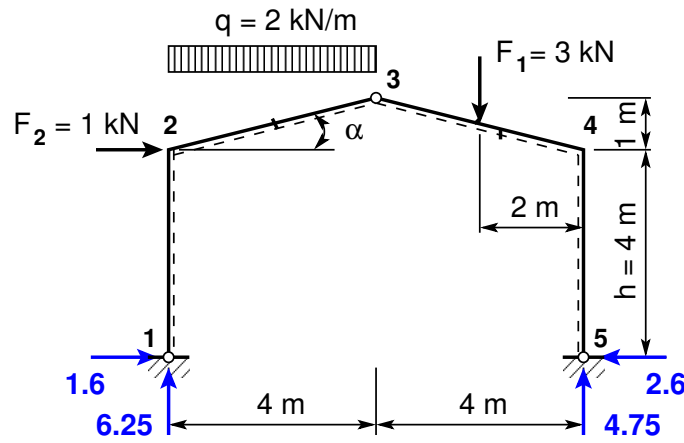


Figure 4.3. Support reactions of the frame EST1

The bending moment, shear force Q and axial force N diagrams of the statically determined three-hinged frame EST1 are shown in Fig. 4.4.

The elements of the frame are depicted in Fig. 4.5 and the sparsity pattern of matrix spA of the frame is shown in Fig. 4.6.

The displacements can be computed with the GNU Octave program `spESTframe3hingeLaheWFL.m`.

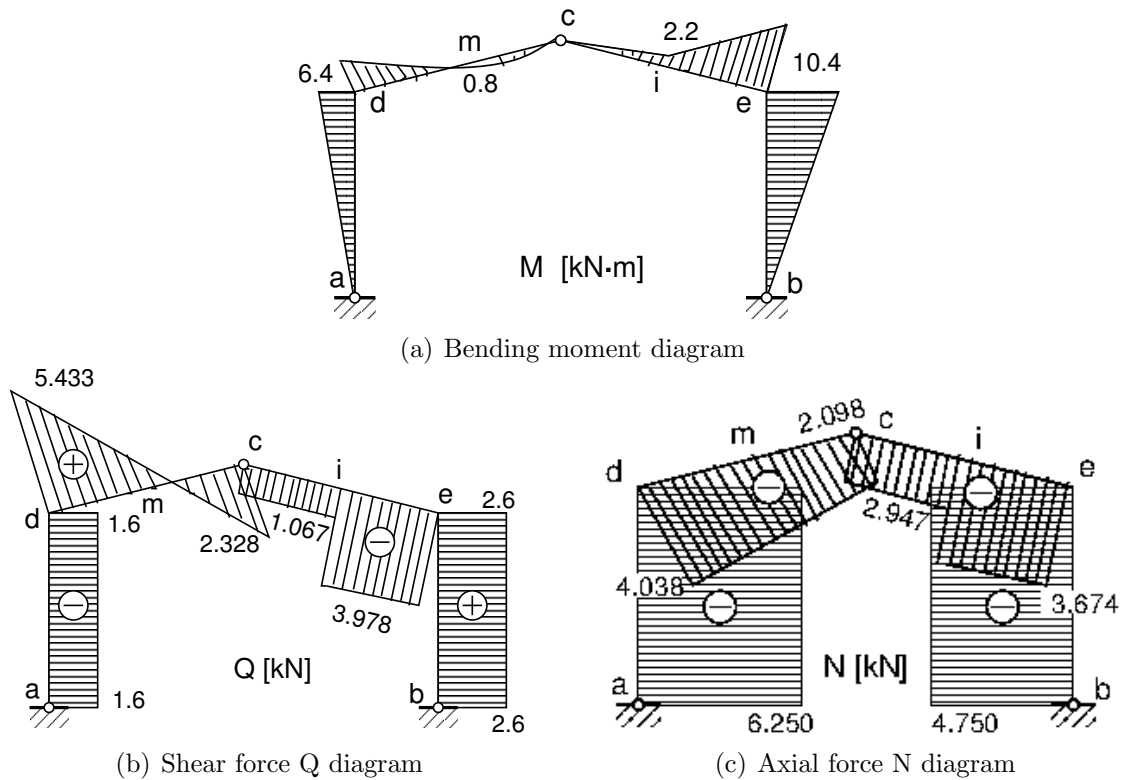


Figure 4.4. Internal forces diagrams of the three-hinged frame EST1

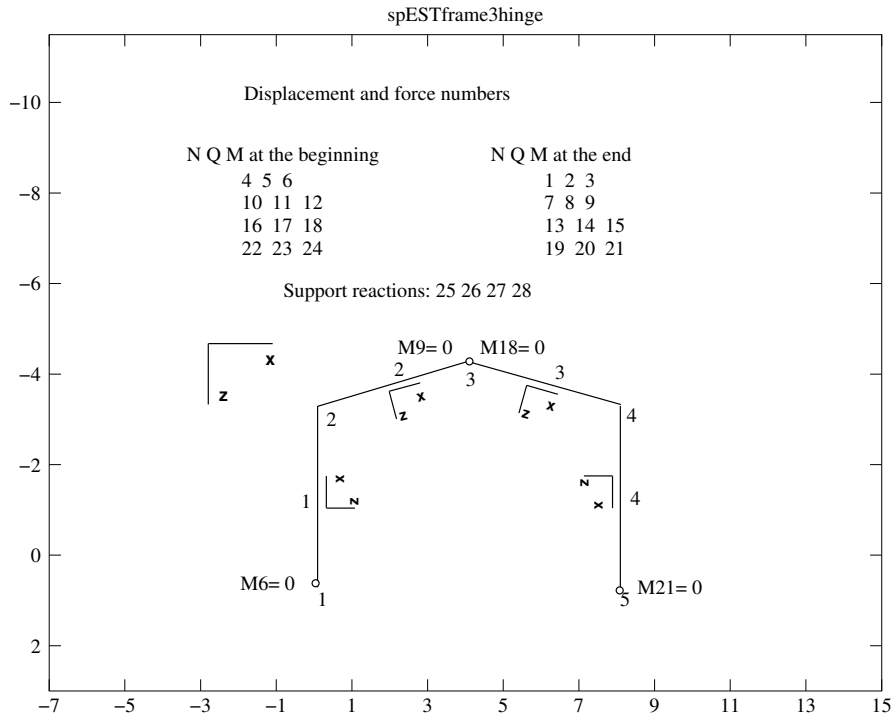


Figure 4.5. Elements of the three-hinged frame EST1

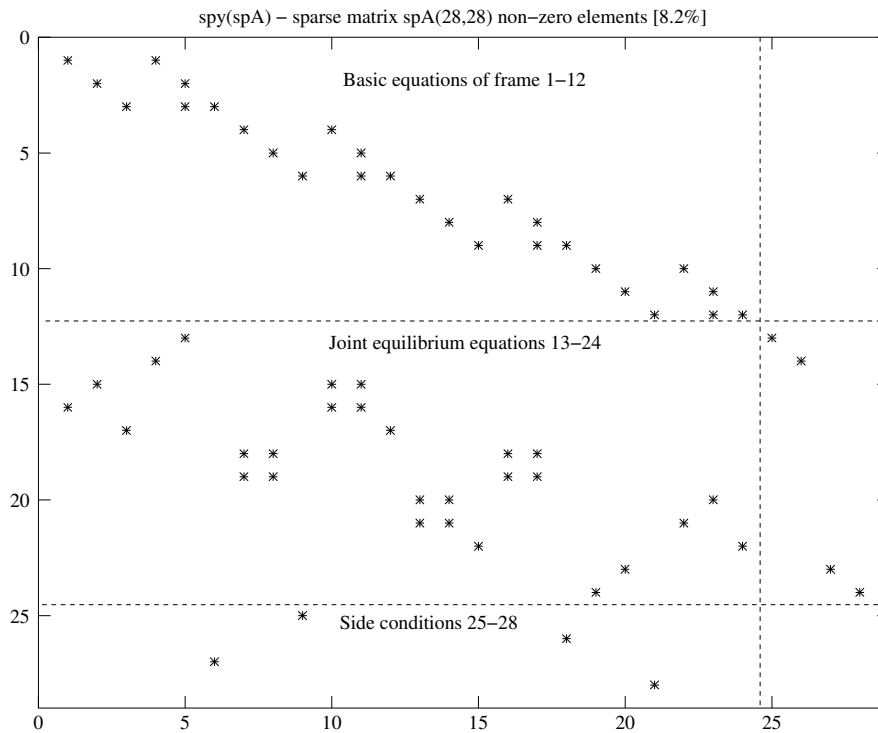


Figure 4.6. Sparsity pattern of matrix spA of the three-hinged frame EST1

4.2 Illustrative beam problem

Example 4.2. Problem Statement. Given is the Gerber beam shown in Fig. 4.7. The beam is loaded with a uniform load $q = 8 \text{ kN/m}$ and with concentrated forces $F_1 = 20 \text{ kN}$, $F_2 = 40 \text{ kN}$, $F_3 = 30 \text{ kN}$, $F_4 = 100 \text{ kN}$, $F_5 = 20 \text{ kN}$.

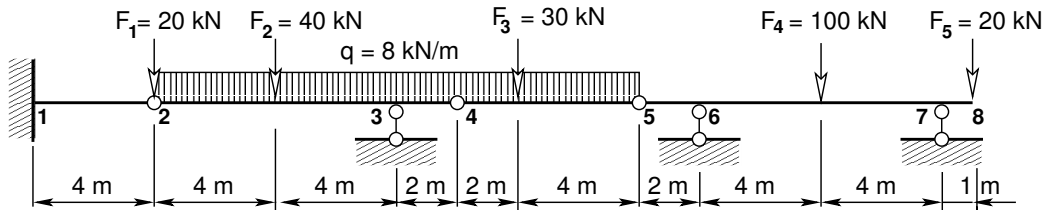


Figure 4.7. The three-span Gerber beam EST

We wish to compute the reactions and internal forces, and draw the shear force and bending moment diagrams.

Problem Solving. To solve the problem, the EST method is used. The solving procedure includes the following.

1. Data input: the number of beam nodes, elements, support reactions; element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (4.5)$$

2. Assembling and solving the boundary problem equations (4.5) (prepared and solved by the program):

- (a) writing the basic equations of beam in transfer matrix form,
- (b) adding the joint equilibrium equations,
- (c) adding the side conditions (hinges),
- (d) solving the system of sparse equations,
- (e) producing an output: initial parameter vectors for element forces; support reactions.

3. Output: element forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program `spESTGerberBeamQM.m` are shown in excerpts from the program: element and nodal loads – excerpt 4.4; nodal coordinates – excerpt 4.5; element properties, topology and hinges – excerpt 4.6.

Program excerpt 4.4 (`spESTGerberBeamQM.m`)

```
Number_of_frame_nodes=8
Number_of_elements=7
Number_of_support_reactions=5
spNNK=4*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
```

```

Forces are calculated on parts ("Nmitmeks") of the element
Nmitmeks=4
Lp=31.0; # graphics axes
# --- Element loads ---

F1=20;                                qz=8.0;
F2=40;                                ## ---- load variants ----
F3=30;                                load_variant=1
F4=100;                               #load_variant=2
F5=20;                                #load_variant=3

switch (koormusvariant)
case{1}
disp(' Load variant 1 ')

#Element load in local coordinates
# qz    qA    qL
# Uniformly distributed load in local coordinate z direction
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,3,ElementideArv);

esQkoormus(1,1:3,1)=[0.0 0.0 4.0];      esQkoormus(1,1:3,5)=[0.0 0.0 2.0];
esQkoormus(1,1:3,2)=[qz 0.0 8.0];      esQkoormus(1,1:3,6)=[0.0 0.0 8.0];
esQkoormus(1,1:3,3)=[qz 0.0 2.0];      esQkoormus(1,1:3,7)=[0.0 0.0 1.0];
esQkoormus(1,1:3,4)=[qz 0.0 6.0];

# Point load in local coordinate z direction kN
# Fz, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:2,1)=[0.0 4.0];              esFjoud(1,1:2,5)=[0.0 2.0];
esFjoud(1,1:2,2)=[F2 4.0];              esFjoud(1,1:2,6)=[F4 4.0];
esFjoud(1,1:2,3)=[0.0 2.0];              esFjoud(1,1:2,7)=[0.0 1.0];
esFjoud(1,1:2,4)=[F3 2.0];

#Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fz; My]
sSolmF = zeros(2,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0;                    s1F(:,1) = sSolmF(:,1,1);
sSolmF(1,1,2)= 20.0;                    s2F(:,1) = sSolmF(:,1,2);
#sSolmF(:,1,3)= 0.0;                    s3F(:,1) = sSolmF(:,1,3);
#sSolmF(:,1,4)= 0.0                     s4F(:,1) = sSolmF(:,1,4);
#sSolmF(:,1,5)= 0.0                     s5F(:,1) = sSolmF(:,1,5);
#sSolmF(:,1,6)= 0.0                     s6F(:,1) = sSolmF(:,1,6);
#sSolmF(:,1,7)= 0.0                     s7F(:,1) = sSolmF(:,1,7);
sSolmF(1,1,8)= 20.0;                    s8F(:,1) = sSolmF(:,1,8);
#%disp(' Kustuta ')

#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force My

```

```

case{2}
disp(' Load variant 2 ')
case{3}
disp(' Load variant 3 ')
#
otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 4.5 (`spESTGerberBeamQM.m`)

```

=====
#      Nodal coordinates
=====
krdn=[#  x
      0.0 ; # node 1
      4.0 ; # node 2
      12.0 ; # node 3
      14.0 ; # node 4
      20.0 ; # node 5
      22.0 ; # node 6
      30.0 ; # node 7
      31.0 ]; # node 8
=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No  w  fi
=====
tsolm=[1  1 1 ; # node 1
       3  1 0 ; # node 2
       6  1 0 ; # node 3
       7  1 0 ]; # node 4
=====

```

Program excerpt 4.6 (`spESTGerberBeamQM.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element
#  n2 - end of the element
#      n1 - beginning of the element
#      N, Q, M - hinges at the end of the element
#      N, Q, M - hinges at the beginning of the element
  2   1   0 1   0 0;   % element 1
  3   2   0 0   0 1;   % element 2
  4   3   0 1   0 0;   % element 3
  5   4   0 1   0 1;   % element 4
  6   5   0 0   0 1;   % element 5
  7   6   0 0   0 0;   % element 6
  8   7   0 0   0 0]; % element 7
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (4.5), carried out by the function `LaheGerberBeamQM(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the forces at element ends of the frame as shown in Fig. 4.8.*

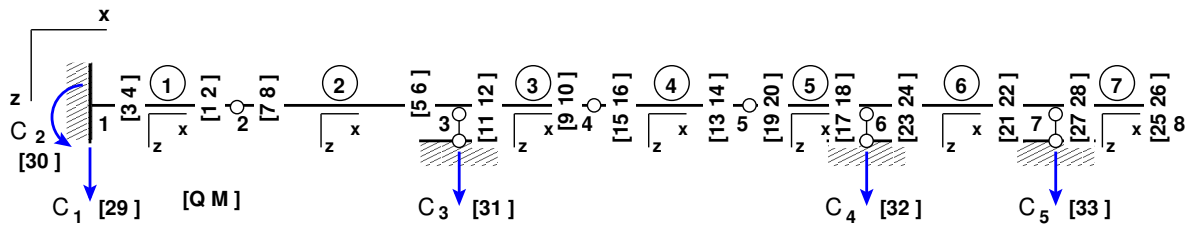


Figure 4.8. Numeration of displacements and forces of the Gerber beam

The initial parameter vectors of the elements are given in excerpt 4.4 from the computing diary. The support reactions of the beam in global coordinates are given in excerpt 4.5 from the computing diary.

Computing diary excerpt 4.4

```
=====
Initial parameter vector
Element No      Q          M
-----
1      -59.000    236.000
2      -39.000     0.000
3      -60.000    104.000
4      -44.000     0.000
5       34.000     0.000
6      -56.000     68.000
7      -20.000     20.000
=====
```

Computing diary excerpt 4.5

```
Support reactions begin from X row: 29
=====
No          X          Node    Cz <=> 1
                                   Cy <=> 2
-----
29    -5.900000e+01    1         1
30     2.360000e+02    1         2
31    -1.250000e+02    3         1
32    -9.000000e+01    6         1
33    -6.400000e+01    7         1
=====
```

Testing a static equilibrium for the Gerber beam EST

Consider next a static equilibrium of the beam shown in Figs. 4.7 and 4.8. Let us project the forces onto the Z-axis:

$$\sum Z = 0; \quad C_1 + C_3 + C_4 + C_5 + F_1 + F_2 + F_3 + F_4 + F_5 + q \cdot 16 = -59.0 - 125.0 - 90.0 - 64.0 + 20.0 + 40.0 + 30.0 + 100.0 + 20.0 + 8.0 \cdot 16 = 0.0 \quad (4.6)$$

We now write equation (4.7) of the sum of the moments and the moments of the forces acting about point 1 shown in Figs. 4.7 and 4.8:

$$\begin{aligned} \sum M_1 = 0; \quad & C_2 - 12.0 \cdot C_3 - 22.0 \cdot C_4 - 30.0 \cdot C_5 - 4.0 \cdot F_1 - 8.0 \cdot F_2 - 16.0 \cdot F_3 \\ & - 26.0 \cdot F_4 - 31.0 \cdot F_5 - 12.0 \cdot 16.0 \cdot q = 236.0 + 12.0 \cdot 125.0 + 22.0 \cdot 90.0 \\ & + 30.0 \cdot 64.0 - 4.0 \cdot 20.0 - 8.0 \cdot 40.0 - 16.0 \cdot 30.0 - 26.0 \cdot 100.0 - 31.0 \cdot 20.0 \\ & - 12.0 \cdot 16.0 \cdot 8.0 = 0.0 \end{aligned} \quad (4.7)$$

The calculations with equations (4.6) and (4.7) have verified the static equilibrium of the Gerber beam EST.

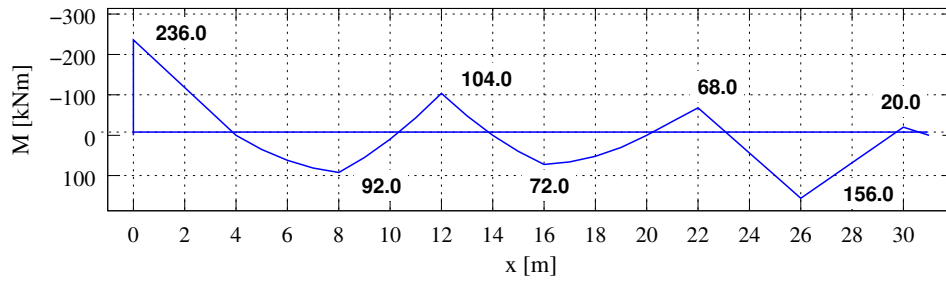
3. Output: the forces determined by the transfer matrix are shown in excerpt 4.6 from the computing diary.

Computing diary excerpt 4.6 (spESTGerberBeamQM.m)

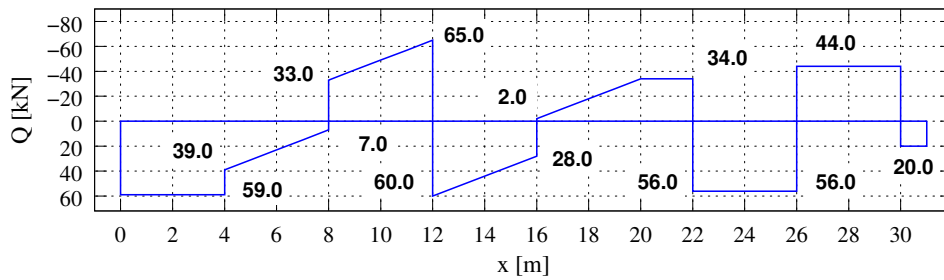
```

=====
Element forces determined by transfer matrix
=====
Forces of element no 1 of length 4.000 m
The element is divided into 4 parts
shear force Q -   59.000   59.000   59.000   59.000   59.000
moment force M - -236.000 -177.000 -118.000 -59.000   0.000
-----
Forces of element no 2 of length 8.000 m
The element is divided into 4 parts
shear force Q -   39.000   23.000  -33.000  -49.000  -65.000
moment force M -    0.000   62.000   92.000   10.000 -104.000
-----
Forces of element no 3 of length 2.000 m
The element is divided into 4 parts
shear force Q -   60.000   56.000   52.000   48.000   44.000
moment force M - -104.000  -75.000  -48.000  -23.000   0.000
-----
Forces of element no 4 of length 6.000 m
The element is divided into 4 parts
shear force Q -   44.000   32.000  -10.000  -22.000  -34.000
moment force M -    0.000   57.000   66.000   42.000   0.000
-----
Forces of element no 5 of length 2.000 m
The element is divided into 4 parts
shear force Q -  -34.000  -34.000  -34.000  -34.000  -34.000
moment force M -    0.000  -17.000  -34.000  -51.000  -68.000
-----
Forces of element no 6 of length 8.000 m
The element is divided into 4 parts
shear force Q -   56.000   56.000  -44.000  -44.000  -44.000
moment force M -  -68.000   44.000  156.000   68.000  -20.000
-----
Forces of element no 7 of length 1.000 m
The element is divided into 4 parts
shear force Q -   20.000   20.000   20.000   20.000   20.000
moment force M -  -20.000  -15.000  -10.000   -5.000   0.000
-----
Forces of element no 4 at x = 1.999999
shear force Q -    28.000
moment force M -    72.000
-----
Forces of element no 2 at x = 3.999999
shear force Q -     7.000
moment force M -    92.000
-----
Forces of element no 6 at x = 3.999999
shear force Q -    56.000
moment force M -   156.000
-----

```

(a) Bending moment diagram



(b) Shear force Q diagram

Figure 4.9. Internal forces diagrams of the Gerber beam EST

The bending moment and shear force Q diagrams of the Gerber beam EST are shown in Fig. 4.9.

In Fig. 4.10, the elements, and in Fig. 4.11, the sparsity pattern of matrix spA of the Gerber beam are shown.

The displacements can be computed with the GNU Octave program *spESTGerber-BeamWFI.m*.

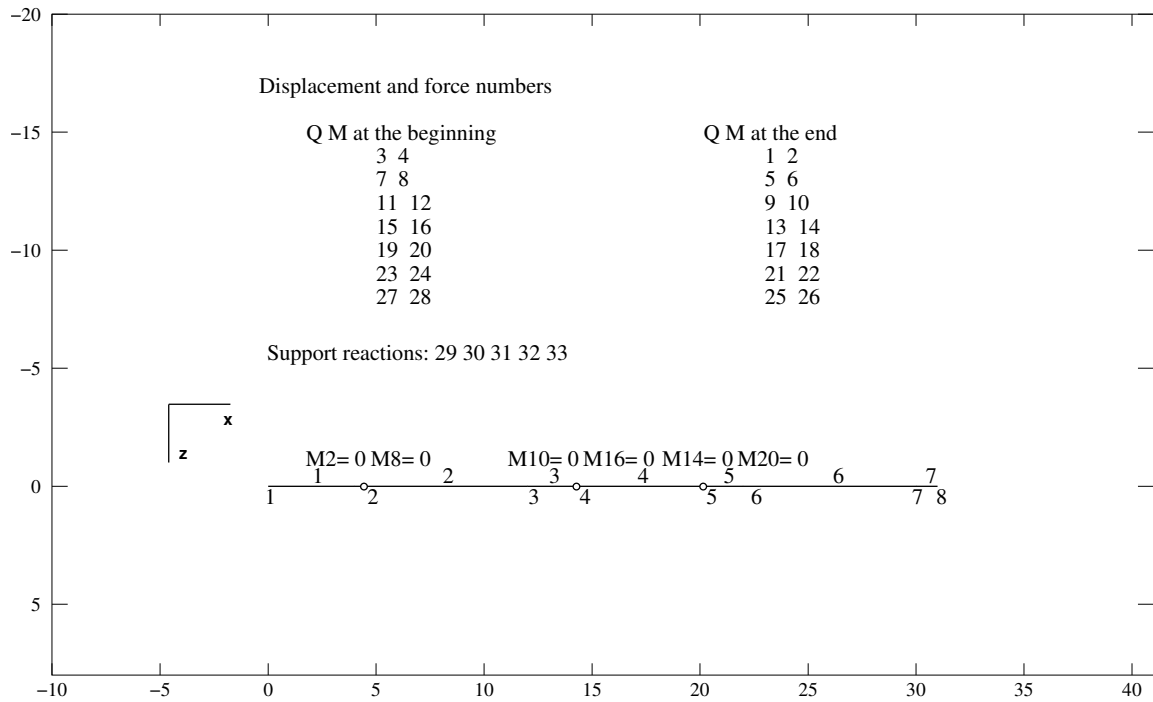


Figure 4.10. Elements of the Gerber beam EST

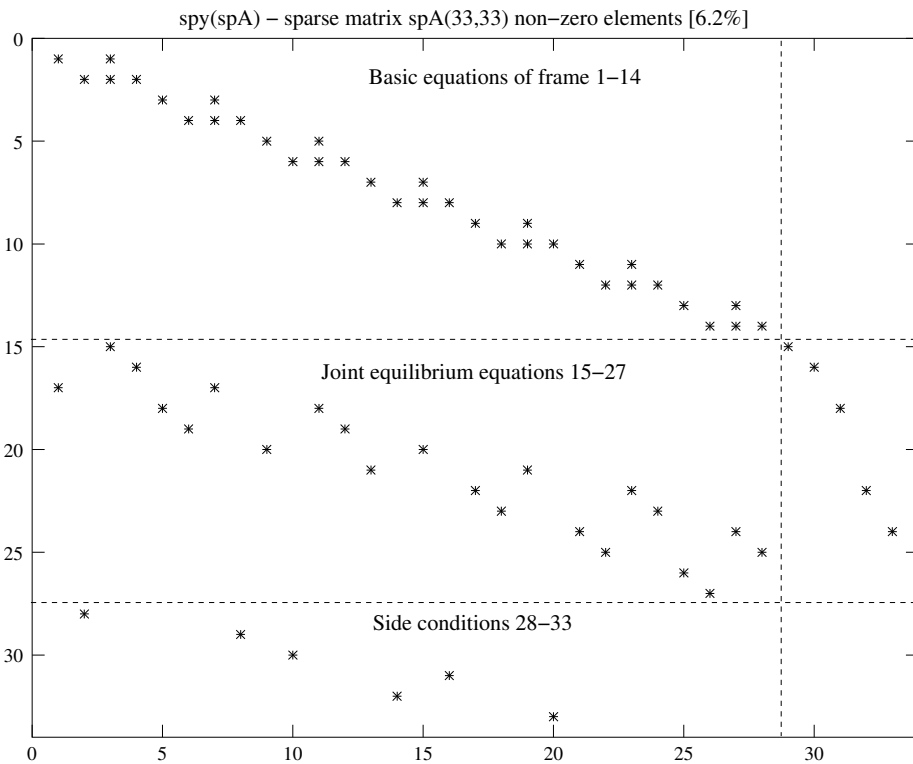


Figure 4.11. Sparsity pattern of matrix spA of the Gerber beam EST

4.3 Illustrative truss problem

Example 4.3. Problem Statement. Determine the forces in the truss shown in Fig. 4.12. The truss of length 24 m, panel length 3 m, and height 4 m is loaded at bottom chord nodes by vertical forces $F_3 = 5$ kN, $F_5 = 10$ kN, $F_7 = 4$ kN, $F_9 = 8$ kN, $F_{13} = 4$ kN, $F_{15} = 8$ kN.

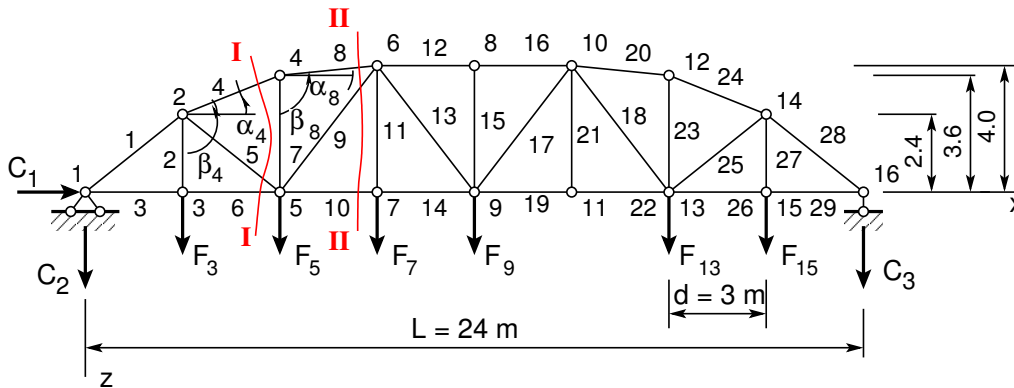


Figure 4.12. A polygonal eight-panel truss

We wish to compute the reactions, internal forces N_7 , N_8 , N_9 , and N_{10} , and draw influence line diagrams for the truss members 7, 8, 9, and 10.

Problem Solving. To solve the problem, the EST method is employed. For assembling and solving the boundary problem equation

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (4.8)$$

the GNU Octave program `spESTtrussN2.m` is used.

1. **Input data** for this program are shown in excerpts from the program: element and nodal loads – excerpt 4.7; nodal coordinates – excerpt 4.8; topology and hinges – excerpt 4.9.

Program excerpt 4.7 (`spESTtrussN2.m`)

```

Number_of_truss_nodes=16
Number_of_elements=29
Number_of_support_reactions=3
spNNK=Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Lp=24.0; # graphics axes
# --- Truss properties ---

d=3.0;      # panel length
Npanel=8;  # number of panels
L=24.0;    # truss span
H=4.0;     # truss height
jaotT=L/8;

# ---- load variants ----
load_variant=1;
#load_variant=2
#load_variant=3

```

```

switch (koormusvariant)
case{1}
disp(' Load variant 1 ')
#
#=====
mjNr1=5; # influence line diagram numbers
mjNr2=2;
mjNr3=4;
mjNr4=7;
mjNr5=9;
#=====

#=====
# Node forces in global coordinates
#=====
VJoud=[# Fx      Fz                Fx      Fz
      0      0;    # node 1        0      8;    # node 9
      0      0;    # node 2        0      0;    # node 10
      0      5;    # node 3        0      0;    # node 11
      0      0;    # node 4        0      0;    # node 12
      0     10;    # node 5        0      4;    # node 13
      0      0;    # node 6        0      0;    # node 14
      0      4;    # node 7        0      8;    # node 15
      0      0;    # node 8        0     0]; # node 16

#=====
# Unit load acting at nodes
#=====
YJoudS=[# Fx      Fz                Fx      Fz
      0      1;    # node 1        0      1;    # node 9
      0      0;    # node 2        0      0;    # node 10
      0      1;    # node 3        0      1;    # node 11
      0      0;    # node 4        0      0;    # node 12
      0      1;    # node 5        0      1;    # node 13
      0      0;    # node 6        0      0;    # node 14
      0      1;    # node 7        0      1;    # node 15
      0      0;    # node 8        0     1]; # node 16

#
case{2}
disp(' Load variant 2 ')
#
case{3}
disp(' Load variant 3 ')
#
otherwise
disp(' No load variant cases ')
endswitch
#

```

Program excerpt 4.8 (`spESTtrussN2.m`)

```

#=====
#      Nodal coordinates
#=====
krdn=[# x          z          x          z

      0.0      0.0;    # node 1          12.0      0.0;    # node 9
      3.0     -2.4;    # node 2          15.0     -4.0;    # node 10
      3.0      0.0;    # node 3          15.0      0.0;    # node 11
      6.0     -3.6;    # node 4          18.0     -3.6;    # node 12
      6.0      0.0;    # node 5          18.0      0.0;    # node 13
      9.0     -4.0;    # node 6          21.0     -2.4;    # node 14
      9.0      0.0;    # node 7          21.0      0.0;    # node 15
      12.0    -4.0;    # node 8          24.0      0.0];   # node 16

#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support    u    w
#=====
tsolm=[# x          z          x          z

      1      1;    # node 1          0      0;    # node 9
      0      0;    # node 2          0      0;    # node 10
      0      0;    # node 3          0      0;    # node 11
      0      0;    # node 4          0      0;    # node 12
      0      0;    # node 5          0      0;    # node 13
      0      0;    # node 6          0      0;    # node 14
      0      0;    # node 7          0      0;    # node 15
      0      0;    # node 8          0      1];   # node 16

```

Program excerpt 4.9 (`spESTtrussN2.m`)

```

#=====
#----- Element topology -----
#=====
selemjl=[# n1 - beginning of the element
#          n2 - end of the element

      1  2;    # element 1          8 10;    # element 16
      2  3;    # element 2          9 10;    # element 17
      1  3;    # element 3          10 13;   # element 18
      2  4;    # element 4          9 11;    # element 19
      2  5;    # element 5          10 12;   # element 20
      3  5;    # element 6          10 11;   # element 21
      4  5;    # element 7          11 13;   # element 22
      4  6;    # element 8          12 13;   # element 23
      5  6;    # element 9          12 14;   # element 24
      5  7;    # element 10         13 14;   # element 25
      6  7;    # element 11         13 15;   # element 26
      6  8;    # element 12         14 15;   # element 27
      6  9;    # element 13         14 16;   # element 28
      7  9;    # element 14         15 16];   # element 29
      8  9;    # element 15

```

The procedure for computing the internal forces in truss members is discussed in [Lah12].

3. **Output:** the internal forces of elements given in excerpt 4.7 from the computing diary.

Computing diary excerpt 4.7 (load case 1 spESTtrussN2.m)

```

=====
Internal forces in truss members
The last 3 forces are support reactions
No      N
-----
1      -32.6157
2       5.0000
3      25.4686
4     -32.0864
5       5.5361
6      25.4686
7       7.9445
8     -30.0552
9      -1.7536
10     30.8438
11      4.0000
12    -31.8750
13      1.7187
14     30.8438
15     -0.0000
16    -31.8750
17      8.2813
18     -4.2189
19     26.9063
20    -24.5906
21      0.0000
22     26.9063
23      6.5000
24    -26.2525
25      1.4008
26     23.2811
27      8.0000
28    -29.8143
29     23.2811
30      0.0000
31    -20.3750
32    -18.6250

```

In excerpt 4.8 from the computing diary, the influence line ordinates are for the members belonging to the lower chord. The procedure for computing the influence line ordinates is discussed in [KW90] and [Lah12].

The influence line diagram of member 9 is shown in Fig. 4.13.

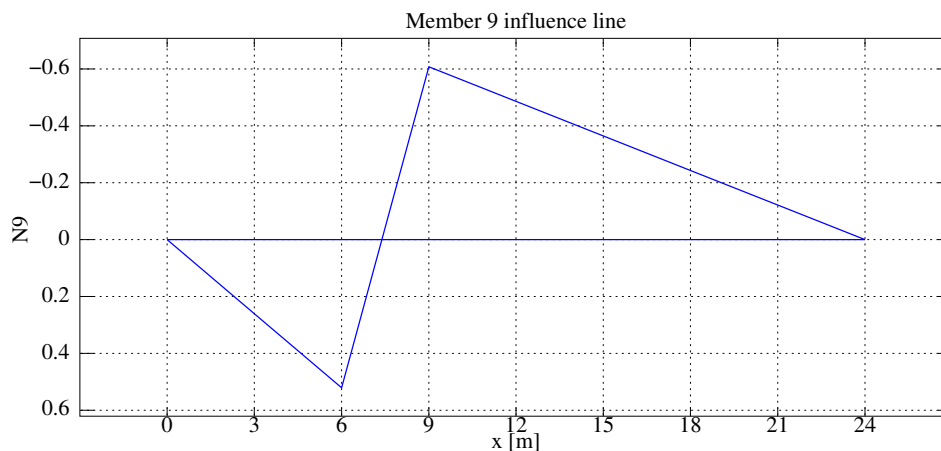


Figure 4.13. Influence line for member 9

Computing diary excerpt 4.8 (`spESTtrussN2.m`)

```

=====
Influence line x-coordinates
-----
      0      3      6      9     12     15     18     21     24
-----
=====

Influence line ordinates
The last 3 rows are for support reactions
-----
  1  0.0000 -1.4007 -1.2006 -1.0005 -0.8004 -0.6003 -0.4002 -0.2001  0.0000
  2  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
  3  0.0000  1.0937  0.9375  0.7812  0.6250  0.4687  0.3125  0.1562  0.0000
  4  0.0000 -0.6731 -1.3463 -1.1219 -0.8975 -0.6731 -0.4488 -0.2244  0.0000
  5  0.0000 -0.6003  0.4002  0.3335  0.2668  0.2001  0.1334  0.0667  0.0000
  6  0.0000  1.0937  0.9375  0.7812  0.6250  0.4687  0.3125  0.1562  0.0000
  7  0.0000  0.1667  0.3333  0.2778  0.2222  0.1667  0.1111  0.0556  0.0000
  8  0.0000 -0.6305 -1.2611 -1.0509 -0.8407 -0.6305 -0.4204 -0.2102  0.0000
  9  0.0000  0.2604  0.5208 -0.6076 -0.4861 -0.3646 -0.2431 -0.1215  0.0000
 10  0.0000  0.4688  0.9375  1.4063  1.1250  0.8438  0.5625  0.2813  0.0000
 11  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 12  0.0000 -0.3750 -0.7500 -1.1250 -1.5000 -1.1250 -0.7500 -0.3750  0.0000
 13 -0.0000 -0.1563 -0.3125 -0.4688  0.6250  0.4688  0.3125  0.1563  0.0000
 14  0.0000  0.4688  0.9375  1.4063  1.1250  0.8438  0.5625  0.2813  0.0000
 15  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 16  0.0000 -0.3750 -0.7500 -1.1250 -1.5000 -1.1250 -0.7500 -0.3750  0.0000
 17  0.0000  0.1563  0.3125  0.4688  0.6250 -0.4688 -0.3125 -0.1563  0.0000
 18  0.0000 -0.1215 -0.2431 -0.3646 -0.4861 -0.6076  0.5208  0.2604  0.0000
 19  0.0000  0.2813  0.5625  0.8438  1.1250  1.4063  0.9375  0.4688  0.0000
 20  0.0000 -0.2102 -0.4204 -0.6305 -0.8407 -1.0509 -1.2611 -0.6305  0.0000
 21  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
 22  0.0000  0.2813  0.5625  0.8438  1.1250  1.4063  0.9375  0.4688  0.0000
 23  0.0000  0.0556  0.1111  0.1667  0.2222  0.2778  0.3333  0.1667  0.0000
 24  0.0000 -0.2244 -0.4488 -0.6731 -0.8975 -1.1219 -1.3463 -0.6731  0.0000
 25  0.0000  0.0667  0.1334  0.2001  0.2668  0.3335  0.4002 -0.6003  0.0000
 26  0.0000  0.1562  0.3125  0.4687  0.6250  0.7812  0.9375  1.0937  0.0000
 27  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
 28  0.0000 -0.2001 -0.4002 -0.6003 -0.8004 -1.0005 -1.2006 -1.4007  0.0000
 29  0.0000  0.1562  0.3125  0.4687  0.6250  0.7812  0.9375  1.0937  0.0000
 30  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
 31 -1.0000 -0.8750 -0.7500 -0.6250 -0.5000 -0.3750 -0.2500 -0.1250  0.0000
 32  0.0000 -0.1250 -0.2500 -0.3750 -0.5000 -0.6250 -0.7500 -0.8750 -1.0000
-----

```

The sparsity pattern of matrix spA of the polygonal truss is given in Fig. 4.14.

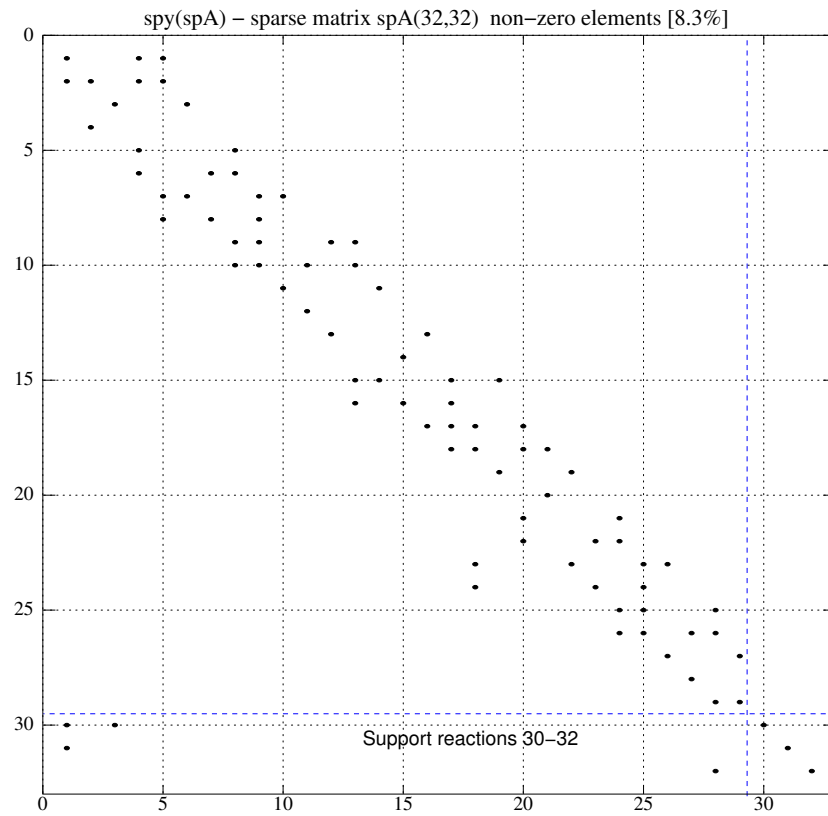


Figure 4.14. Sparsity pattern of matrix spA of the polygonal truss

Part II

Second-order structural analysis

5. Second-order structural analysis

5.1 Introduction

A beam-column is a kind of beam subjected to both lateral loads and axial compression simultaneously. Additional moments developed by an axial force can also be of importance. This effect of the axial force is referred to as the *beam-column effect* or *P-delta effect* (P- Δ effect). Secondary moments from the compressive axial load may have a deleterious effect on the strength and stability of a beam-column. Deflections, bending moments, and shear forces in a beam-column are not proportional to the applied loads. Thus the superposition of deflections cannot be applied. Equilibrium equations are based on the deformed geometry of the structure. An analysis of a beam-column is a nonlinear problem. In an analysis of frames, axial forces are generally not known at the outset. Accordingly, a set of axial forces of the members is determined as a linear structure. At the second iteration, the axial forces from the first iteration are used. If the results show that the axial forces obtained by the second iteration differ greatly from those of the first iteration, the calculated values will be used to find new values, and the analysis is repeated.

5.2 The governing equation for a beam-column

Consider the deformed beam-column element in Fig. 5.1. We must differentiate the forces in the deformed and undeformed axis of the beam. The axial force S versus normal force N and the transverse shear force H versus shear force Q are depicted in Fig. 5.1.

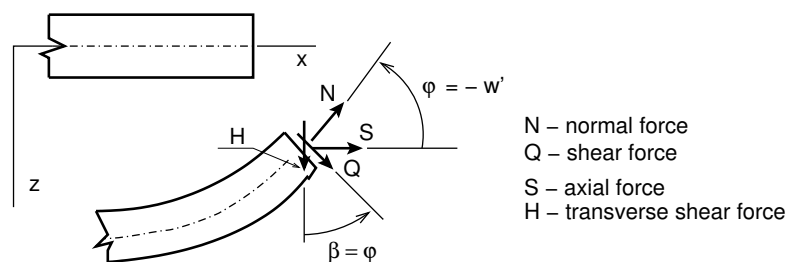


Figure 5.1. Normal, shear, axial and transverse shear forces

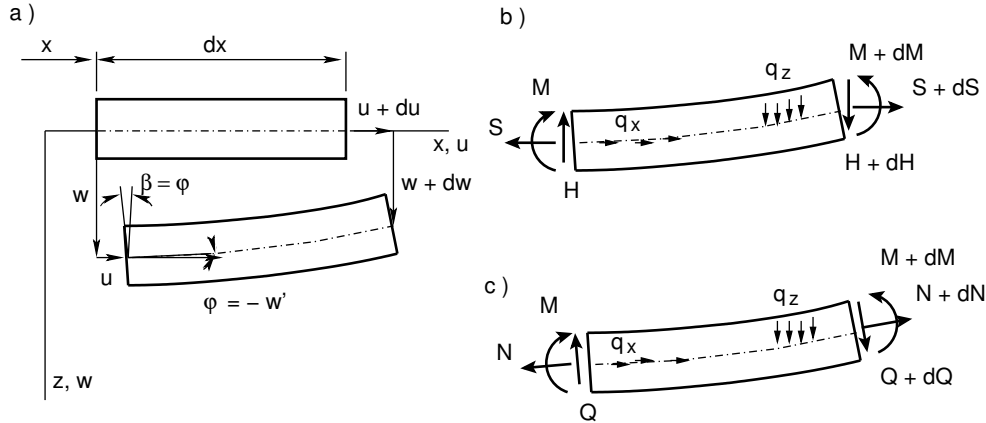


Figure 5.2. Deformed beam elements with the axial and normal forces

The beam-column theory uses the Bernoulli¹-Euler² theory kinematic assumption that the curvature ψ_y can be considered equal to the second derivative of the deflected longitudinal axis w'' .

$$\psi_y = \frac{-w''}{(1 + w'^2)^{3/2}} \approx -w'' \quad (5.1)$$

The relationship of Eq. (A.26) between the forces referred to the deformed and undeformed axis is

$$\begin{bmatrix} S \\ H \end{bmatrix} = \begin{bmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{bmatrix} \begin{bmatrix} N \\ Q \end{bmatrix} \quad (5.2)$$

The “small slope” assumption, $\varphi \ll 1$, allows us to conclude that

$$\cos \varphi \approx 1, \quad \sin \varphi \approx \tan \varphi \approx \varphi = -w' \quad (5.3)$$

and the relationship between the forces referred to the deformed and undeformed axis can be rewritten according to [RH95] and [PW94] in the form

$$\begin{bmatrix} S \\ H \end{bmatrix} = \begin{bmatrix} 1 & \varphi \\ -\varphi & 1 \end{bmatrix} \begin{bmatrix} N \\ Q \end{bmatrix} \quad (5.4)$$

Now we consider Fig. 5.2 c, and project the forces onto the *direction of z-axis*:

$$\begin{aligned} \Sigma Z = 0 : \quad & N \underbrace{\sin \varphi}_{\approx \varphi} - Q \underbrace{\cos \varphi}_{\approx 1} + q_z dx \\ & - (N + dN) \underbrace{\sin(\varphi + d\varphi)}_{\approx \varphi + d\varphi} + (Q + dQ) \underbrace{\cos(\varphi + d\varphi)}_{\approx 1} = 0 \end{aligned} \quad (5.5)$$

¹Jacob Bernoulli (1655–1705), a Swiss mathematician.

²Leonhard Euler (1707–1783), a Swiss mathematician and physicist.

Taking into account the assumption of Eq. (5.3), we rewrite Eq. (5.5):

$$\begin{aligned} \Sigma Z = 0 : \quad & N\varphi - Q + q_z dx \\ & -N\varphi - Nd\varphi - dN\varphi - \underbrace{dNd\varphi}_{\approx 0} + Q + dQ = 0 \end{aligned} \quad (5.6)$$

After reducing Eq. (5.6),

$$-N \frac{d\varphi}{dx} - \varphi \frac{dN}{dx} + \frac{dQ}{dx} + q_z(x) = 0 \quad (5.7)$$

or

$$\frac{d}{dx} (-N\varphi + Q) + q_z(x) = 0 \quad (5.8)$$

From Eq. (5.4) we have

$$-N\varphi + Q = H \quad (5.9)$$

Using the previous equation, we can rewrite the equilibrium equation (5.8) in the form

$$\frac{d}{dx} H + q_z(x) = 0 \quad (5.10)$$

Now we project the forces onto *the direction of x-axis*:

$$\begin{aligned} \Sigma X = 0 : \quad & -N \underbrace{\cos \varphi}_{\approx 1} - Q \underbrace{\sin \varphi}_{\approx \varphi} + q_x dx \\ & + (N + dN) \underbrace{\cos(\varphi + d\varphi)}_{\approx 1} + (Q + dQ) \underbrace{\sin(\varphi + d\varphi)}_{\approx \varphi + d\varphi} = 0 \end{aligned} \quad (5.11)$$

With the assumption of Eq. (5.3),

$$\begin{aligned} \Sigma X = 0 : \quad & -N - Q\varphi + q_x dx \\ & + N + dN + Q\varphi + dQ\varphi + \varphi dQ + \underbrace{dQd\varphi}_{\approx 0} = 0 \end{aligned} \quad (5.12)$$

Reducing this equation gives

$$\frac{d}{dx} N + Q \frac{d}{dx} \varphi + \varphi \frac{d}{dx} Q + q_x(x) = 0 \quad (5.13)$$

or

$$\frac{d}{dx} (N + Q\varphi) + q_x(x) = 0 \quad (5.14)$$

From Eq. (5.4)

$$N + \underbrace{Q\varphi}_{\ll N} = S_x \cong N \quad (5.15)$$

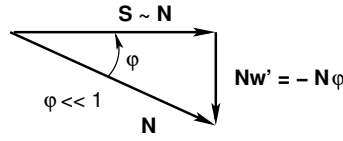


Figure 5.3. Components of the normal force

Using Eqs. (5.15) and (5.14), we get

$$\frac{d}{dx}S_x + q_x(x) = 0 \quad (5.16)$$

Next we write the equation of the sum of the moments and the moments of the forces acting about the end point of the element shown in Fig. 5.2 c:

$$\Sigma M_y = 0; \quad M + dM - M - Qds = 0 \quad (5.17)$$

Here, ds is the length of the element. Hence $ds \approx dx$, and we rewrite Eq. (5.17):

$$\frac{d}{dx}M - Q = 0 \quad (5.18)$$

Due to the orthogonality of Eq. (A.27), we can write an inverse relationship to Eq. (5.4)

$$\begin{bmatrix} N \\ Q \end{bmatrix} = \begin{bmatrix} 1 & -\varphi \\ \varphi & 1 \end{bmatrix} \begin{bmatrix} S \\ H \end{bmatrix} \quad (5.19)$$

where the expression for the shear force Q is

$$Q = S\varphi + H \quad (5.20)$$

Replacing Q from Eq. (5.18) by the expression (5.20), we obtain

$$\frac{d}{dx}M - S\varphi - H = 0 \quad (5.21)$$

Hence $S_x \cong N$, and from Eq. (5.16)

$$N' + q_x(x) = 0 \quad (5.22)$$

Differentiating Eq. (5.21) we have

$$M'' - (S_x\varphi)' - H' = 0 \quad (5.23)$$

Replacing H' in the above equation by the expression from Eq. (5.10), and φ by the expression $\varphi = \beta = -w'$, we obtain

$$M'' + (S_x w')' + q_z(x) = 0 \quad (5.24)$$

or

$$M'' + S'_x w' + (S_x w')' + S_x w' + q_z(x) = 0 \tag{5.25}$$

For a constant axial force S_x and a constant EI_y we obtain

$$M'' + S_x w'' + q_z(x) = 0 \tag{5.26}$$

$$M = -EI_y w'' \tag{5.27}$$

The governing differential equations take the form

$$M'' \mp \left| \frac{S_x}{EI_y} \right| M + q_z(x) = 0 \tag{5.28}$$

where the minus sign denotes a *tensile* axial force ($S \geq 0$) and the plus sign a *compressive* axial force ($S \leq 0$).

The governing equations (5.28) for the beam-column transverse displacement w can be written as

$$[EI_y w'''] \mp |S_x| w'' - q_z(x) = 0 \tag{5.29}$$

and for a constant EI_y , we obtain

$$[w'''] \mp \left| \frac{S_x}{EI_y} \right| w'' - \frac{q_z(x)}{EI_y} = 0 \tag{5.30}$$

where the minus sign marks a *tensile* axial force ($S \geq 0$) and the plus sign a *compressive* axial force ($S \leq 0$).

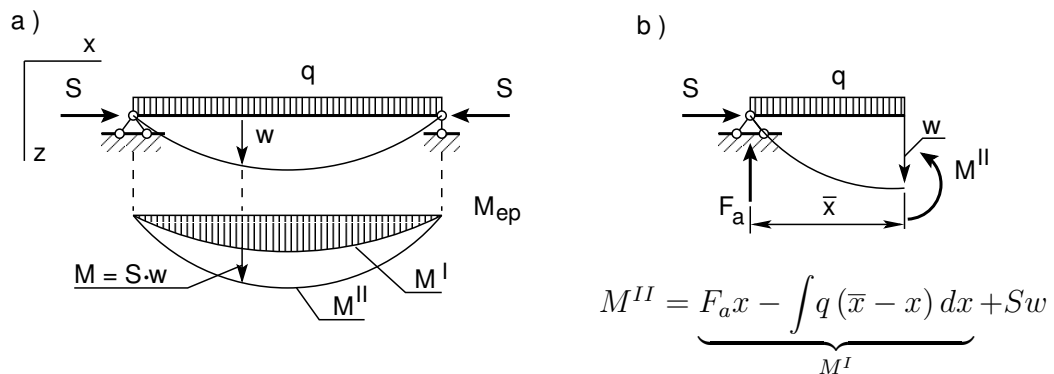


Figure 5.4. Bending moments of a beam-column

Equation (5.30) is the differential equation governing the deflection w of a beam-column member with a constant EI_y , subjected to a *constant axial force* at any end restraint.

In a second-order analysis, the total deflection w is calculated (see Fig. 5.4). *The total bending moment M^{II}* is the sum of the bending moment M^I of undeformed geometry of the member and the moment Sw (S – axial load, w – displacement) due to the deformed geometry of the member shown in Fig. 5.4 b. The moment is said to be amplified.

$$M^{II} = M^I + S_x w \quad (5.31)$$

The axial force S can be presented with the displacements u and w :

$$S \approx N = -EA \left[u' + \frac{1}{2} (w')^2 \right] \quad (5.32)$$

where $\left[u' + \frac{1}{2} (w')^2 \right]$ is a large Kármán³-type deflection and EA is the axial stiffness.

5.3 Solutions of governing equations

We shall now apply the techniques of subsection 1.2.1 to establish the transfer matrix for a beam element subjected to an axial force S . The differential equation for a compressive axial force will be obtained from Eq. (5.30):

$$\frac{d^4 w}{dx^4} + \left| \frac{S}{EI_y} \right| \frac{d^2 w}{dx^2} + \frac{q(x)}{EI_y} = 0 \quad (5.33)$$

For a tensile axial force

$$\frac{d^4 w}{dx^4} - \left| \frac{S}{EI_y} \right| \frac{d^2 w}{dx^2} + \frac{q(x)}{EI_y} = 0 \quad (5.34)$$

We will be looking for the general solution of the non-homogeneous differential equation (5.33) in the form

$$w = w_0 w_1 + w_0' w_2 + w_0'' w_3 + w_0''' w_4 + w_e(x) \quad (5.35)$$

where w_0, w_0', w_0'', w_0''' are parameter values of the searchable function at $x = x_0$; w_1, w_2, w_3, w_4 are a *normed fundamental set* to the associated homogeneous differential equation, and $w_e(x)$ is the particular solution of the non-homogeneous differential equation.

³Theodore von Kármán (1881–1963), a Hungarian-American mathematician, aerospace engineer and physicist.

5.3.1 The set of solutions to a homogeneous differential equation

The fundamental set of solutions to the homogeneous differential equation

$$\frac{d^4 w}{dx^4} \pm \left| \frac{S}{EI_y} \right| \frac{d^2 w}{dx^2} = 0 \quad (5.36)$$

for a compressive axial force with magnitude $|S|$ is

$$w_1^* = 1, \quad w_2^* = \frac{\nu}{l}x, \quad w_3^* = \cos\left(\frac{\nu}{l}x\right), \quad w_4^* = \sin\left(\frac{\nu}{l}x\right) \quad (5.37)$$

and for a tensile axial force with magnitude $|S|$

$$w_1^* = 1, \quad w_2^* = \frac{\nu}{l}x, \quad w_3^* = \operatorname{ch}\left(\frac{\nu}{l}x\right), \quad w_4^* = \operatorname{sh}\left(\frac{\nu}{l}x\right) \quad (5.38)$$

where $\nu = l\sqrt{S/EI_y}$ is a characteristic parameter of a beam-column member and l is the length of the member.

Consider next the Wronski determinant $W(x)$ of the fundamental set of solutions from Eq. (5.37):

$$W(x) = \begin{vmatrix} 1 & \frac{\nu}{l}x & \cos\left(\frac{\nu}{l}x\right) & a \sin\left(\frac{\nu}{l}x\right) \\ 0 & \frac{\nu}{l} & -\frac{\nu}{l} \sin\left(\frac{\nu}{l}x\right) & \frac{\nu}{l} \cos\left(\frac{\nu}{l}x\right) \\ 0 & 0 & -\left(\frac{\nu}{l}\right)^2 \cos\left(\frac{\nu}{l}x\right) & -\left(\frac{\nu}{l}\right)^2 \sin\left(\frac{\nu}{l}x\right) \\ 0 & 0 & \left(\frac{\nu}{l}\right)^3 \sin\left(\frac{\nu}{l}x\right) & -\left(\frac{\nu}{l}\right)^3 \cos\left(\frac{\nu}{l}x\right) \end{vmatrix} \quad (5.39)$$

Hence the Wronskian $W(x)$ value at $x = 0$ is not 1:

$$W(0) = \begin{vmatrix} 1 & 0 & 1 & 0 \\ 0 & \frac{\nu}{l} & 0 & \frac{\nu}{l} \\ 0 & 0 & -\left(\frac{\nu}{l}\right)^2 & 0 \\ 0 & 0 & 0 & -\left(\frac{\nu}{l}\right)^3 \end{vmatrix} \neq 1 \quad (5.40)$$

Let us norm the Wronskian of the previous equation by

- subtracting the 1st column from the 3rd column and multiplying the result by $\left(-\left(\frac{l}{\nu}\right)^2\right)$
- subtracting the 2nd column from the 4th column and multiplying the result by $\left(-\left(\frac{l}{\nu}\right)^3\right)$
- multiplying the 2nd column by $\left(\frac{l}{\nu}\right)$.

Now the value of the Wronskian

$$W(x=0) = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1 \quad (5.41)$$

We make similar rearrangements with the fundamental set of solutions from Eq. (5.37) (with columns of Eq. (5.39)). Once more we

- subtract the 1st column from the 3rd column and multiply the result by $\left(-\left(\frac{l}{\nu}\right)^2\right)$:
 $-\left(\frac{l}{\nu}\right)^2 \left[\cos\left(\frac{\nu}{l}x\right) - 1\right]$
- subtract the 2nd column from the 4th column and multiply the result by $\left(-\left(\frac{l}{\nu}\right)^3\right)$:
 $-\left(\frac{l}{\nu}\right)^3 \left[\sin\left(\frac{\nu}{l}x\right) - \left(\frac{\nu}{l}x\right)\right]$
- multiply the 2nd column by $\left(\frac{l}{\nu}\right)$: x .

We get the *normed fundamental set of solutions for the compressive axial force*:

$$\begin{aligned} w_1 &= 1, & w_2 &= x, \\ w_3 &= -\left(\frac{l}{\nu}\right)^2 \left[\cos\left(\frac{\nu}{l}x\right) - 1\right], & w_4 &= -\left(\frac{l}{\nu}\right)^3 \left[\sin\left(\frac{\nu}{l}x\right) - \left(\frac{\nu}{l}x\right)\right] \end{aligned} \quad (5.42)$$

The *normed fundamental set of solutions for the tensile axial force* is:

$$\begin{aligned} w_1 &= 1, & w_2 &= x, \\ w_3 &= \left(\frac{l}{\nu}\right)^2 \left[\operatorname{ch}\left(\frac{\nu}{l}x\right) - 1\right], & w_4 &= \left(\frac{l}{\nu}\right)^3 \left[\operatorname{sh}\left(\frac{\nu}{l}x\right) - \left(\frac{\nu}{l}x\right)\right] \end{aligned} \quad (5.43)$$

There are two sign conventions (see Fig. (1.2)) for the internal reactions⁴⁵ (contact forces).

For the parameters w_0 , w'_0 , w''_0 , and w'''_0 of the searchable function at $x = x_0$ (Sign Convention 1) we obtain

$$w_0 = w_0, \quad w'_0 = -\varphi_0, \quad w''_0 = -\frac{M_y}{EI_y}, \quad w'''_0 = -\frac{Q_z}{EI_y} \quad (5.44)$$

and for Sign Convention 2

$$w_0 = w_0, \quad w'_0 = -\varphi_0, \quad w''_0 = \frac{M_y}{EI_y}, \quad w'''_0 = \frac{Q_z}{EI_y} \quad (5.45)$$

⁴... cut the frame into its component members and find *the internal reactions* [MPHS].

⁵The fixed-end forces and moments at joints are called *the internal reactions* [WP960].

The complete solution for the compressive axial force (Sign Convention 1) is

$$w = w_0 - \varphi_0 x + \frac{M_y}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^2 \left[\cos \left(\frac{\nu}{l} x \right) - 1 \right] + \frac{Q_z}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^3 \left[\sin \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] + w_e(x) \quad (5.46)$$

and for the tensile axial force (Sign Convention 1)

$$w = w_0 - \varphi_0 x - \frac{M_y}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^2 \left[\operatorname{ch} \left(\frac{\nu}{l} x \right) - 1 \right] - \frac{Q_z}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^3 \left[\operatorname{sh} \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] + w_e(x) \quad (5.47)$$

The complete solution for the compressive axial force (Sign Convention 2) is

$$w = w_0 - \varphi_0 x - \frac{M_y}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^2 \left[\cos \left(\frac{\nu}{l} x \right) - 1 \right] - \frac{Q_z}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^3 \left[\sin \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] + w_e(x) \quad (5.48)$$

and for the tensile axial force (Sign Convention 2)

$$w = w_0 - \varphi_0 x + \frac{M_y}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^2 \left[\operatorname{ch} \left(\frac{\nu}{l} x \right) - 1 \right] + \frac{Q_z}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^3 \left[\operatorname{sh} \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] + w_e(x) \quad (5.49)$$

Relations between the 1st and 2nd sign conventions for a bending moment and shear force:

$$\begin{aligned} M_y \Big|_{o(\text{Sign Convention 1})} &= -M_y \Big|_{o(\text{Sign Convention 2})} \\ Q_z \Big|_{o(\text{Sign Convention 1})} &= -Q_z \Big|_{o(\text{Sign Convention 2})} \end{aligned}$$

5.3.2 Transfer matrix for a beam element with axial force

Differentiating the solution $w(x)$ of Eq. (5.46) to a homogeneous differential equation for the compressive axial force (Sign Convention 1)

$$w(x) = w_0 - x\varphi_0 + \left[\sin \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] \left(\frac{l}{\nu} \right)^3 \frac{Q_z \Big|_o}{EI_y} + \left[\cos \left(\frac{\nu}{l} x \right) - 1 \right] \left(\frac{l}{\nu} \right)^2 \frac{M_y \Big|_o}{EI_y} \quad (5.50)$$

we get the expression for rotation:

$$\begin{aligned} \varphi_x = -w'_x(x) = 0 \cdot w_0 - 1 \cdot \varphi_0 - \left[\cos\left(\frac{\nu}{l}x\right) - 1 \right] \left(\frac{l}{\nu}\right)^2 \frac{Q_z|_0}{EI_y} \\ + \left[\sin\left(\frac{\nu}{l}x\right) \right] \left(\frac{l}{\nu}\right) \frac{M_y|_0}{EI_y} \end{aligned} \quad (5.51)$$

Next we take the second derivative to find the bending moment $M_y(x)$:

$$\begin{aligned} M_y(x) = -EI_y w''_x(x) = 0 \cdot w_0 - 0 \cdot \varphi_0 + \left[\sin\left(\frac{\nu}{l}x\right) \right] \left(\frac{l}{\nu}\right) Q_z|_0 \\ + \left[\cos\left(\frac{\nu}{l}x\right) \right] M_y|_0 \end{aligned} \quad (5.52)$$

Taking a derivative from the bending moment $M_y(x)$, we obtain the shear force $Q_z(x)$:

$$\begin{aligned} Q_z(x) = -EI_y w'''_x(x) = 0 \cdot w_0 - 0 \cdot \varphi_0 + \left[\cos\left(\frac{\nu}{l}x\right) \right] Q_z|_0 \\ - \left[\sin\left(\frac{\nu}{l}x\right) \right] \frac{l}{\nu} M_y|_0 \end{aligned} \quad (5.53)$$

For the constant normal force $N_x(x)$ and axial displacement $u(x)$ we take (Sign Convention 1)

$$u(x) = u|_0 + \frac{x}{NA} N_x|_0 \quad (5.54)$$

$$N_x(x) = N_x|_0 \quad (5.55)$$

Now consider the vectors

$$\mathbf{Z}_{\mathbf{x}}^{(\mathbf{N})} = \begin{bmatrix} u(x) \\ w(x) \\ \varphi_y(x) \\ N_x(x) \\ Q_z(x) \\ M_y(x) \end{bmatrix}_{\mathbf{x}}, \quad \mathbf{Z}_{\mathbf{A}}^{(\mathbf{N})} = \begin{bmatrix} u_0 \\ w_0 \\ \varphi_0 \\ N_x|_0 \\ Q_z|_0 \\ M_y|_0 \end{bmatrix}_{\mathbf{A}} \quad (5.56)$$

the elements of which can be found from Eqs. (5.50)–(5.55).

These equations can now be written in matrix form:

$$\mathbf{Z}_{\mathbf{x}}^{(\mathbf{N})} = \mathbf{U}^{(\mathbf{N}-1)} \cdot \mathbf{Z}_{\mathbf{A}}^{(\mathbf{N})} \quad (5.57)$$

where $\mathbf{U}^{(\mathbf{N}-1)}$ is the transfer matrix for the compressive axial force (Sign Convention 1).

$$\mathbf{U}^{(\mathbf{N}-1)} = \begin{bmatrix} 1 & 0 & 0 & \frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & -\left[\nu \frac{x}{l} - \sin\left(\nu \frac{x}{l}\right)\right] \frac{l^3}{\nu^3 EI_y} & -\left[1 - \cos\left(\nu \frac{x}{l}\right)\right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & \left[1 - \cos\left(\nu \frac{x}{l}\right)\right] \frac{l^2}{\nu^2 EI_y} & \sin\left(\nu \frac{x}{l}\right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos\left(\nu \frac{x}{l}\right) & -\sin\left(\nu \frac{x}{l}\right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & \sin\left(\nu \frac{x}{l}\right) \frac{l}{\nu} & \cos\left(\nu \frac{x}{l}\right) \end{bmatrix} \quad (5.58)$$

For the tensile axial force (Sign Convention 1), the transfer equations are given by

$$\mathbf{Z}_x^{(N)} = \mathbf{U}^{(N+1)} \cdot \mathbf{Z}_A^{(N)} \quad (5.59)$$

where $\mathbf{U}^{(N+1)}$ is the transfer matrix:

$$\mathbf{U}^{(N+1)} = \begin{bmatrix} 1 & 0 & 0 & \frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & \left[\nu \frac{x}{l} - \text{sh} \left(\nu \frac{x}{l} \right) \right] \frac{l^3}{\nu^3 EI_y} & \left[1 - \text{ch} \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & - \left[1 - \text{ch} \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} & \text{sh} \left(\nu \frac{x}{l} \right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{ch} \left(\nu \frac{x}{l} \right) & \text{sh} \left(\nu \frac{x}{l} \right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & \text{sh} \left(\nu \frac{x}{l} \right) \frac{l}{\nu} & \text{ch} \left(\nu \frac{x}{l} \right) \end{bmatrix} \quad (5.60)$$

For the compressive axial force (Sign Convention 2), the transfer equations are given by

$$\mathbf{Z}_x^{(N)} = \mathbf{U}^{(N-2)} \cdot \mathbf{Z}_A^{(N)} \quad (5.61)$$

where $\mathbf{U}^{(N-2)}$ is the transfer matrix:

$$\mathbf{U}^{(N-2)} = \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & \left[\nu \frac{x}{l} - \sin \left(\nu \frac{x}{l} \right) \right] \frac{l^3}{\nu^3 EI_y} & \left[1 - \cos \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & - \left[1 - \cos \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} & - \sin \left(\nu \frac{x}{l} \right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & - \cos \left(\nu \frac{x}{l} \right) & \sin \left(\nu \frac{x}{l} \right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & - \sin \left(\nu \frac{x}{l} \right) \frac{l}{\nu} & - \cos \left(\nu \frac{x}{l} \right) \end{bmatrix} \quad (5.62)$$

This transfer matrix has been obtained from Eqs. (5.58) and (5.60) by multiplying the 4th, 5th and 6th columns by -1 (see Sign Convention, Fig. 1.2).

For the tensile axial force (Sign Convention 2), the transfer equations are given by

$$\mathbf{Z}_x^{(N)} = \mathbf{U}^{(N+2)} \cdot \mathbf{Z}_A^{(N)} \quad (5.63)$$

where $\mathbf{U}^{(N+2)}$ is the transfer matrix:

$$\mathbf{U}^{(N+2)} = \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & - \left[\nu \frac{x}{l} - \text{sh} \left(\nu \frac{x}{l} \right) \right] \frac{l^3}{\nu^3 EI_y} & - \left[1 - \text{ch} \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & \left[1 - \text{ch} \left(\nu \frac{x}{l} \right) \right] \frac{l^2}{\nu^2 EI_y} & - \text{sh} \left(\nu \frac{x}{l} \right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & - \text{ch} \left(\nu \frac{x}{l} \right) & - \text{sh} \left(\nu \frac{x}{l} \right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & - \text{sh} \left(\nu \frac{x}{l} \right) \frac{l}{\nu} & - \text{ch} \left(\nu \frac{x}{l} \right) \end{bmatrix} \quad (5.64)$$

The transformation matrices of Eqs. (5.62) and (5.64) can be represented with the GNU Octave function `y1fmII.m` (p. 249).

5.3.3 The particular solution of a differential equation

The particular solution $w_e(x)$ (see Eq. (5.35)) of the non-homogeneous differential equation

$$[w'''] \mp \left| \frac{S_x}{EI_y} \right| w'' = q_z/EI_y + \mathcal{F}_z \delta(t - x_0)/EI_y + \mathcal{M}_x \delta'(t - x_0)/EI_y \quad (5.65)$$

can be obtained with the Cauchy⁶ formula [Stp59]:

$$w_e(x) = \int_{x_0}^x K(x, t) f_n(t) dt \quad (5.66)$$

where $K(x, t)$ is the normed solution of the corresponding homogeneous differential equation. To be more precise,

$$w_e(x) = \int_{x_0}^x K_4(x, t) f_4(t) dt + \int_{x_0}^x K_3(x, t) f_3(t) dt + \int_{x_0}^x K_2(x, t) f_2(t) dt \quad (5.67)$$

Using the *normed fundamental set of solutions for the compressive axial force* from Eq. (5.42), we obtain

$$K_4(x, t) = w_4(x - t) = - \left(\frac{l}{\nu} \right)^3 \left[\sin \left(\frac{\nu}{l} (x - t) \right) - \left(\frac{\nu}{l} (x - t) \right) \right] \quad (5.68)$$

$$K_3(x, t) = w_3(x - t) = - \left(\frac{l}{\nu} \right)^2 \left[\cos \left(\frac{\nu}{l} (x - t) \right) - 1 \right] \quad (5.69)$$

$$K_2(x, t) = w_2(x - t) = (x - t) \quad (5.70)$$

and the load functions $f_n(t)$ are

$$f_4(t) = q_z(t)/EI_y, \quad f_3(t) = \mathcal{F}_z/EI_y, \quad f_2(t) = \mathcal{M}_x/EI_y \quad (5.71)$$

For the particular solution $w_e(x)$ at the constant load q_z , we evaluate the following integral:

$$w_e(x) = - \left(\frac{l}{\nu} \right)^3 \frac{q_z}{EI_y} \int_{x_0}^x \left[\sin \left(\frac{\nu}{l} (x - t) \right) - \left(\frac{\nu}{l} (x - t) \right) \right] dt \quad (5.72)$$

Let us start with the first term of Eq. (5.72):

$$\int_{x_0}^x \sin \left(\frac{\nu}{l} (x - t) \right) dt = \frac{l}{\nu} \left[1 - \cos \left(\frac{\nu}{l} (x - x_0) \right) \right] \quad (5.73)$$

For the second term of the equation (5.72) we get

$$\int_{x_0}^x \frac{\nu}{l} (x - t) dt = \frac{l}{\nu} \frac{\nu^2 (x - x_0)^2}{l^2} \frac{1}{2} \quad (5.74)$$

⁶Augustin-Louis Cauchy (1789–1857), a French mathematician.

At the constant load q_z , the particular solution is

$$w_e(x) = \frac{1}{2\nu^4} \left[\frac{\nu^2(x-x_0)^2}{l^2} - 2 \left(1 - \cos \frac{\nu}{l}(x-x_0) \right) \right] \frac{q_z l^4}{EI_y} \quad (5.75)$$

If we take $x-x_0=l$, this particular solution will be in agreement with those examined in [Krä91a], [Krä91b], and [Bor79b].

In case of the point load \mathcal{F}_z , the particular solution is

$$\begin{aligned} w_e(x) &= - \left(\frac{l}{\nu} \right)^2 \frac{\mathcal{F}_z}{EI_y} \int_{x_0}^x \left[\cos \left(\frac{\nu}{l}(x-t) \right) - 1 \right] dt = \\ &= - \left[\sin \left(\frac{\nu}{l}(x-x_0) \right) - \left(\frac{\nu}{l}(x-x_0) \right) \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y} \end{aligned} \quad (5.76)$$

The general solution of the non-homogeneous differential equation (5.65) is the sum of the solution of the homogeneous differential equation (5.48) (Sign Convention 2) and the particular solutions of Eqs. (5.75), (5.76), and (1.36).

$$\begin{aligned} w = w_0 - \varphi_0 x &- \frac{Q_z}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^3 \left[\sin \left(\frac{\nu}{l}x \right) - \left(\frac{\nu}{l}x \right) \right] - \frac{M_y}{EI_y} \Big|_o \left(\frac{l}{\nu} \right)^2 \left[\cos \left(\frac{\nu}{l}x \right) - 1 \right] \\ &+ \frac{1}{2\nu^4} \left[\frac{\nu^2(x-x_0)^2}{l^2} - 2 \left(1 - \cos \frac{\nu(x-x_0)}{l} \right) \right] \frac{q_z l^4}{EI_y} \\ &+ \left[\left(\frac{\nu}{l}(x-x_0) \right) - \sin \left(\frac{\nu}{l}(x-x_0) \right) \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y} \end{aligned} \quad (5.77)$$

Let us take the derivatives of the displacement of Eq. (5.77) and apply these to the governing differential equations (5.3), (5.27), and (5.18).

We get the following beam governing equations in transfer matrix form (for the compressive axial force, Sign Convention 2):

$$\underbrace{\begin{bmatrix} w_x \\ \varphi_x \\ \dots \\ Q_x \\ M_x \end{bmatrix}}_{\mathbf{z}_x} = \underbrace{\begin{bmatrix} 1 & -x & \vdots & \frac{l^3}{\nu^3 EI_y} \left[\frac{\nu x}{l} - \sin \frac{\nu x}{l} \right] & \frac{l^2}{\nu^2 EI_y} \left[1 - \cos \frac{\nu x}{l} \right] \\ 0 & 1 & \vdots & -\frac{l^2}{\nu^2 EI_y} \left[1 - \cos \frac{\nu x}{l} \right] & -\frac{l}{\nu EI_y} \sin \frac{\nu x}{l} \\ \dots & \dots & \vdots & \dots & \dots \\ 0 & 0 & \vdots & -\cos \frac{\nu x}{l} & \frac{\nu}{l} \sin \frac{\nu x}{l} \\ 0 & 0 & \vdots & -\frac{l}{\nu} \sin \frac{\nu x}{l} & -\cos \frac{\nu x}{l} \end{bmatrix}}_{\mathbf{U}_x^{(c)}} \underbrace{\begin{bmatrix} w_A \\ \varphi_A \\ \dots \\ Q_A \\ M_A \end{bmatrix}}_{\mathbf{z}_A} + \underbrace{\begin{bmatrix} \frac{1}{2\nu^4} \left[\frac{\nu^2(x-x_0)^2}{l^2} - 2 \left(1 - \cos \frac{\nu(x-x_0)}{l} \right) \right] \frac{q_z l^4}{EI_y} \\ - \left[\frac{\nu(x-x_0)}{l} - \sin \frac{\nu(x-x_0)}{l} \right] \frac{q l^3}{\nu^3 EI_y} \\ \dots \\ - \left[\sin \frac{\nu(x-x_0)}{l} \right] \frac{q l}{\nu} \\ - \left[1 - \cos \frac{\nu(x-x_0)}{l} \right] \frac{q l^2}{\nu^2} \end{bmatrix}}_{\mathbf{z} \dots}$$

$$+ \underbrace{\begin{bmatrix} \left[\frac{\nu(x-x_0)}{l} - \sin \frac{\nu(x-x_0)}{l} \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y} \\ - \left[1 - \cos \frac{\nu(x-x_0)}{l} \right] \frac{\mathcal{F}_z l^2}{\nu^2 EI_y} \\ \dots \\ - \left[\cos \frac{\nu(x-x_0)}{l} \right] \mathcal{F}_z \\ - \left[\sin \frac{\nu(x-x_0)}{l} \right] \frac{\mathcal{F}_z l}{\nu} \end{bmatrix}}{\dots \overset{\circ}{\mathbf{Z}}} \quad (5.78)$$

Employing a symbolic matrix notation, the above equations can be written as

$$\mathbf{Z}_x = \mathbf{U}_x^{(C)} \mathbf{Z}_A + \overset{\circ}{\mathbf{Z}} \quad (5.79)$$

Consider next the finding of the particular solution with the Cauchy formula of Eq. (5.66) where the *normed fundamental set of solutions for the tensile axial force* of Eq. (5.43) is

$$K_4(x, t) = w_4(x - t) = \left(\frac{l}{\nu} \right)^3 \left[\text{sh} \left(\frac{\nu}{l} (x - t) \right) - \left(\frac{\nu}{l} (x - t) \right) \right] \quad (5.80)$$

$$K_3(x, t) = w_3(x - t) = \left(\frac{l}{\nu} \right)^2 \left[\text{ch} \left(\frac{\nu}{l} (x - t) \right) - 1 \right] \quad (5.81)$$

$$K_2(x, t) = w_2(x - t) = (x - t) \quad (5.82)$$

The load functions are

$$f_4(t) = q_z(t)/EI_y, \quad f_3(t) = \mathcal{F}_z/EI_y, \quad f_2(t) = \mathcal{M}_x/EI_y \quad (5.83)$$

To obtain the particular solution $w_e(x)$ at the constant load q_z (for the tensile axial force), we evaluate the following integral:

$$w_e(x) = \left(\frac{l}{\nu} \right)^3 \frac{q_z}{EI_y} \int_{x_0}^x \left[\text{sh} \left(\frac{\nu}{l} (x - t) \right) - \left(\frac{\nu}{l} (x - t) \right) \right] dt = \\ - \left[1 + \frac{1}{2} \left(\frac{\nu(x-x_0)}{l} \right)^2 - \text{ch} \frac{\nu(x-x_0)}{l} \right] \frac{ql^4}{\nu^4 EI_y} \quad (5.84)$$

In case of the point load \mathcal{F}_z , the particular solution is

$$w_e(x) = - \left(\frac{l}{\nu} \right)^2 \frac{\mathcal{F}_z}{EI_y} \int_{x_0}^x \left[\text{ch} \left(\frac{\nu}{l} (x - t) \right) - 1 \right] dt = \\ = \left[\frac{\nu}{l} (x - x_0) - \text{sh} \frac{\nu}{l} (x - x_0) \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y} \quad (5.85)$$

The general solution of the non-homogeneous differential equation (5.65) in case of the tensile axial force is the sum of the solution of the homogeneous differential equation

(5.49) (Sign Convention 2) and the particular solutions of Eqs. (5.84), (5.85), and (1.36).

$$\begin{aligned}
w = w_0 - \varphi_0 x &+ \frac{Q_z}{EI_y} \Big|_0 \left(\frac{l}{\nu} \right)^3 \left[\text{sh} \left(\frac{\nu}{l} x \right) - \left(\frac{\nu}{l} x \right) \right] + \frac{M_y}{EI_y} \Big|_0 \left(\frac{l}{\nu} \right)^2 \left[\text{ch} \left(\frac{\nu}{l} x \right) - 1 \right] \\
&- \left[1 + \frac{1}{2} \left(\frac{\nu(x-x_0)}{l} \right)^2 - \text{ch} \frac{\nu(x-x_0)}{l} \right] \frac{ql^4}{\nu^4 EI_y} \\
&+ \left[\frac{\nu}{l} (x-x_0) - \text{sh} \frac{\nu}{l} (x-x_0) \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y}
\end{aligned} \tag{5.86}$$

Let us take the derivatives of the displacement of Eq. (5.86) and apply these to the governing differential equations (5.3), (5.27), and (5.18).

We get the beam governing equations (5.87) in transfer matrix form (for the tensile axial force, Sign Convention 2):

$$\begin{aligned}
\underbrace{\begin{bmatrix} w_x \\ \varphi_x \\ \dots \\ Q_x \\ M_x \end{bmatrix}}_{\mathbf{Z}_x} &= \underbrace{\begin{bmatrix} 1 & -x & \vdots & \frac{l^3}{\nu^3 EI_y} \left[\text{sh} \frac{\nu x}{l} - \frac{\nu x}{l} \right] & \frac{l^2}{\nu^2 EI_y} \left[\text{ch} \frac{\nu x}{l} - 1 \right] \\ 0 & 1 & \vdots & \frac{l^2}{\nu^2 EI_y} \left[1 - \text{ch} \frac{\nu x}{l} \right] & -\frac{l}{\nu EI} \text{sh} \frac{\nu x}{l} \\ \dots & \dots & \vdots & \dots & \dots \\ 0 & 0 & \vdots & -\text{ch} \frac{\nu x}{l} & -\frac{\nu}{l} \text{sh} \frac{\nu x}{l} \\ 0 & 0 & \vdots & -\frac{l}{\nu} \text{sh} \frac{\nu x}{l} & -\text{ch} \frac{\nu x}{l} \end{bmatrix}}_{\mathbf{U}_x^{(T)}} \underbrace{\begin{bmatrix} w_A \\ \varphi_A \\ \dots \\ Q_A \\ M_A \end{bmatrix}}_{\mathbf{Z}_A} \\
&+ \underbrace{\begin{bmatrix} - \left[1 + \frac{1}{2} \left(\frac{\nu(x-x_0)}{l} \right)^2 - \text{ch} \frac{\nu(x-x_0)}{l} \right] \frac{ql^4}{\nu^4 EI_y} \\ \left[\frac{\nu(x-x_0)}{l} - \text{sh} \frac{\nu(x-x_0)}{l} \right] \frac{ql^3}{\nu^3 EI_y} \\ \dots \\ - \left[\text{sh} \frac{\nu(x-x_0)}{l} \right] \frac{ql}{\nu} \\ \left[1 - \text{ch} \frac{\nu(x-x_0)}{l} \right] \frac{ql^2}{\nu^2} \end{bmatrix}}_{\overset{\circ}{\mathbf{Z}} \dots} \\
&+ \underbrace{\begin{bmatrix} \left[\frac{\nu(x-x_0)}{l} - \text{sh} \frac{\nu(x-x_0)}{l} \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI_y} \\ \left[\text{ch} \frac{\nu(x-x_0)}{l} - 1 \right] \frac{\mathcal{F}_z l^2}{\nu^2 EI_y} \\ \dots \\ - \left[\cos \frac{\nu}{l} (x-x_0) \right] \mathcal{F}_z \\ \left[\text{sh} \frac{\nu(x-x_0)}{l} \right] \frac{\mathcal{F}_z l}{\nu} \end{bmatrix}}_{\dots \overset{\circ}{\mathbf{Z}}}
\end{aligned} \tag{5.87}$$

In a symbolic matrix notation, these equations can be written as

$$\mathbf{Z}_x = \mathbf{U}_x^{(T)} \mathbf{Z}_A + \overset{\circ}{\mathbf{Z}} \tag{5.88}$$

Equations (5.78) and (5.87) should be complemented with the normal force N and the displacement u . The axial deformation of a frame element

$$u_x(x) = \frac{x}{EA} N|_0 \quad \text{and} \quad N(x) = N|_0 \quad (5.89)$$

where $N|_0$ – normal force at the beginning of the element,
 EA – axial stiffness of the element.

The symbolic matrix transfer equation with the normal force N and Sign Convention 2 is

$$\mathbf{Z}_x = \mathbf{U}_x \cdot \mathbf{Z}_0 + \overset{\circ}{\mathbf{Z}} \quad (5.90)$$

where

$$\mathbf{Z}_x = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_x, \quad \mathbf{Z}_0 = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_0 \quad (5.91)$$

and

- if the normal force N stands for compression (Sign Convention 2), then the transfer matrix $\mathbf{U}_x \equiv \mathbf{U}_x^{\mathbf{N}-2}$ is that given in Eq. (5.62) and the vector of applied loads, $\overset{\circ}{\mathbf{Z}}^+$, is shown in Table C.5;
- if the normal force N stands for tension (Sign Convention 2), then the transfer matrix $\mathbf{U}_x \equiv \mathbf{U}_x^{\mathbf{N}+2}$ is that given in Eq. (5.64) and the vector of applied loads, $\overset{\circ}{\mathbf{Z}}^-$, is shown in Table C.6.

The transfer matrices $\mathbf{U}_x^{\mathbf{N}-2}$ and $\mathbf{U}_x^{\mathbf{N}+2}$ [Krä91b] can be computed using the GNU Octave function `yIfmII.m` (p. 249).

The loading vectors $\overset{\circ}{\mathbf{Z}}^+$ (5.92) and $\overset{\circ}{\mathbf{Z}}^-$ (5.93) for the constant load q_z in case of the compression and tension force N , respectively, are:

$$\overset{\circ}{\mathbf{Z}}^+ = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ N_e \\ Q_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ \left[\frac{1}{2} \left(\nu \frac{(x-x_0)}{l} \right)^2 - 1 + \cos \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^4}{\nu^4 EI} \\ \left[\sin \left(\nu \frac{(x-x_0)}{l} \right) - \nu \frac{(x-x_0)}{l} \right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ - \left[\sin \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql}{\nu I} \\ \left[\cos \left(\nu \frac{(x-x_0)}{l} \right) - 1 \right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (5.92)$$

$$\overset{\circ}{\mathbf{Z}}^- = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ N_e \\ Q_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ - \left[\frac{1}{2} \left(\nu \frac{(x-x_0)}{l} \right)^2 + 1 - \operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^4}{\nu^4 EI} \\ \left[\nu \frac{(x-x_0)}{l} - \operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ - \left[\operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql}{\nu I} \\ \left[1 - \operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (5.93)$$

The loading vectors can be computed using the GNU Octave function `ylqII.m` (p. 249).

The loading vectors $\overset{\circ}{\mathbf{Z}}^+$ (5.94) and $\overset{\circ}{\mathbf{Z}}^-$ (5.95) for the point load \mathcal{F}_z in case of the compression and tension force N , respectively, are:

$$\overset{\circ}{\mathbf{Z}}^+ = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ N_e \\ Q_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ - \left[\sin \left(\nu \frac{(x-x_0)}{l} \right) - \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{\mathcal{F}_z l^3}{\nu^3 EI} \\ \left[\cos \left(\nu \frac{(x-x_0)}{l} \right) - 1 \right] \frac{\mathcal{F}_z l^2}{\nu^2 EI} \\ 0 \\ - \left[\cos \left(\nu \frac{(x-x_0)}{l} \right) \right] \mathcal{F}_z \\ - \left[\sin \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{\mathcal{F}_z l}{\nu} \end{bmatrix} \quad (5.94)$$

$$\overset{\circ}{\mathbf{Z}}^- = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ N_e \\ Q_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ \left[\operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) - \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl^3}{\nu^3 EI} \\ - \left[\operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) - 1 \right] \frac{Fl^2}{\nu^2 EI} \\ 0 \\ - \left[\operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] F \\ - \left[\operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl}{\nu} \end{bmatrix} \quad (5.95)$$

The loading vectors can be computed using the GNU Octave function `ylffzII.m` (p. 249).

5.3.4 Transformation of a transfer matrix

Let X, Z be the *global coordinate system* (see Fig. 5.5), x, z the *local (initial) coordinate system*, and ξ, η the *current coordinate system* [Yaw09].

We have derived the transfer matrix \mathbf{U}_x from Eq. (5.90) for the normal force N and shear force Q shown in Fig. 5.1. These forces (N, Q) we bind with the *current coordinate system* (ξ, η).

The axial force S and transverse shear force H are described in the *initial (local) coordinate system* (x, z). In the *global coordinate system* (X, Z), the joint equilibrium equations are written.

We now consider a symbolic matrix transfer equation at the axial force S and Sign Convention 2:

$$\mathbf{Z}_x^S = \mathbf{U}_x^S \cdot \mathbf{Z}_0^S + \overset{\circ}{\mathbf{Z}}^{S\pm 2} \quad (5.96)$$

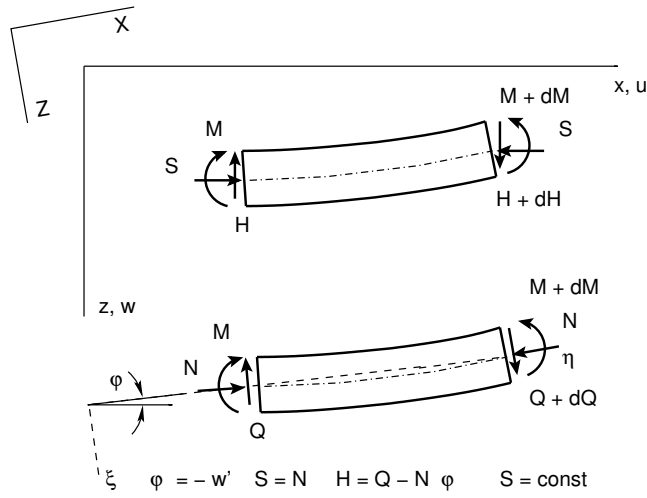


Figure 5.5. Axial and normal forces

where

$$\mathbf{Z}_x^S = \begin{bmatrix} u \\ w \\ \varphi_y \\ S_x \\ H_z \\ M_y \end{bmatrix}_x, \quad \mathbf{Z}_0^S = \begin{bmatrix} u \\ w \\ \varphi_y \\ S_x \\ H_z \\ M_y \end{bmatrix}_0 \quad (5.97)$$

From Eq. (5.4)

$$\begin{aligned} N + \underbrace{Q\varphi}_{\ll N} &= S_x \cong N \\ N &= \mp |S| = \mp \nu^2 \frac{EI}{l^2}, \quad \text{if } \begin{cases} - & \text{then compression} \\ + & \text{then tension} \end{cases} \end{aligned} \quad (5.98)$$

and the relationship between the forces referred to a deformed and undeformed axis can be rewritten [Krä91b] in the form of Eq. (5.4):

$$\begin{bmatrix} H \end{bmatrix} = \begin{bmatrix} -\varphi & 1 \end{bmatrix} \begin{bmatrix} \mp \nu^2 \frac{EI}{l^2} \\ Q \end{bmatrix} = \begin{bmatrix} \pm \nu^2 \frac{EI}{l^2} \varphi & Q \end{bmatrix}, \quad \text{if } \begin{cases} + & \text{then compression} \\ - & \text{then tension} \end{cases} \quad (5.99)$$

Due to the orthogonality of Eq. (A.27), we can write from the inverse of the matrix of Eq. (5.4)

$$\begin{bmatrix} Q \end{bmatrix} = \begin{bmatrix} 1 & \varphi \end{bmatrix} \begin{bmatrix} \mp \nu^2 \frac{EI}{l^2} \\ H \end{bmatrix} = \begin{bmatrix} \mp \nu^2 \frac{EI}{l^2} \varphi & Q \end{bmatrix}, \quad \text{if } \begin{cases} - & \text{then compression} \\ + & \text{then tension} \end{cases} \quad (5.100)$$

In order to change variables N, Q to S, H , we will need the transformation relations of Eq. (5.101) [Krä91b]:

$$\underbrace{\begin{bmatrix} u \\ w \\ \varphi_y \\ S \\ H \\ M_y \end{bmatrix}}_{\mathbf{Z}^S} = \underbrace{\begin{bmatrix} 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \\ . & . & 1 & . & . & . \\ . & . & . & 1 & . & . \\ . & . & \pm \nu^2 \frac{EI}{I^2} & . & 1 & . \\ . & . & . & . & . & 1 \end{bmatrix}}_{\mathbf{T}^{SN}} \underbrace{\begin{bmatrix} u \\ w \\ \varphi_y \\ N \\ Q \\ M_y \end{bmatrix}}_{\mathbf{Z}^N}, \quad \text{if } \begin{cases} + \text{ then compression} \\ - \text{ then tension} \end{cases} \quad (5.101)$$

or

$$\mathbf{Z}^S = \mathbf{T}^{SN} \mathbf{Z}^N \quad (5.102)$$

To compute the transformation matrix \mathbf{T}^{SN} , the GNU Octave function `ytransf.m` (p. 250) can be used.

We will make the inverse change of variables S, H to N, Q with the equations

$$\underbrace{\begin{bmatrix} u \\ w \\ \varphi_y \\ N \\ Q \\ M_y \end{bmatrix}}_{\mathbf{Z}^N} = \underbrace{\begin{bmatrix} 1 & . & . & . & . & . \\ . & 1 & . & . & . & . \\ . & . & 1 & . & . & . \\ . & . & . & 1 & . & . \\ . & . & \mp \nu^2 \frac{EI}{I^2} & . & 1 & . \\ . & . & . & . & . & 1 \end{bmatrix}}_{\mathbf{T}^{NS}} \underbrace{\begin{bmatrix} u \\ w \\ \varphi_y \\ S \\ H \\ M_y \end{bmatrix}}_{\mathbf{Z}^S}, \quad \text{if } \begin{cases} - \text{ then compression} \\ + \text{ then tension} \end{cases} \quad (5.103)$$

or

$$\mathbf{Z}^N = \mathbf{T}^{NS} \mathbf{Z}^S \quad (5.104)$$

Here, the transformation matrix \mathbf{T}^{NS} can be computed using the GNU Octave function `ytransfp.m` (p. 250).

The matrix \mathbf{Z}^N has the inverse matrix \mathbf{Z}^S , i.e

$$\mathbf{Z}^N \cdot \mathbf{Z}^S = I, \quad \mathbf{Z}^S \cdot \mathbf{Z}^N = I \quad (5.105)$$

We rewrite Eq. (5.90) in the form

$$\mathbf{Z}_x^N = \mathbf{U}_x^{N\pm 2} \cdot \underbrace{\mathbf{T}^{NS} \mathbf{Z}_0^S}_{\mathbf{z}_0^N} + \overset{\circ}{\mathbf{Z}}^{N\pm 2} \quad (5.106)$$

and multiply Eq. (5.106) from left by \mathbf{T}^{SN} :

$$\underbrace{\mathbf{T}^{SN} \cdot \mathbf{Z}_x^N}_{\mathbf{z}_x^S} = \underbrace{\mathbf{T}^{SN} \cdot \mathbf{U}_x^{N\pm 2} \cdot \mathbf{T}^{NS}}_{\mathbf{U}_x^{S\pm 2}} \mathbf{Z}_0^S + \underbrace{\mathbf{T}^{SN} \cdot \overset{\circ}{\mathbf{Z}}^{N\pm 2}}_{\overset{\circ}{\mathbf{z}}^{S\pm 2}} \quad (5.107)$$

When comparing equations (5.96) and (5.107), we show that

$$\mathbf{U}_x^{\mathbf{S}\pm 2} = \mathbf{T}^{\mathbf{SN}} \cdot \mathbf{U}_x^{\mathbf{N}\pm 2} \cdot \mathbf{T}^{\mathbf{NS}} \quad (5.108)$$

and

$$\overset{\circ}{\mathbf{Z}}^{S\pm 2} = \mathbf{T}^{\mathbf{SN}} \cdot \overset{\circ}{\mathbf{Z}}^{N\pm 2} \quad (5.109)$$

where $\mathbf{U}_x^{\mathbf{S}-2} =$

$$= \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -\sin\left(\frac{\nu x}{l}\right) \frac{l}{\nu} & 0 & \left[\frac{\nu x}{l} - \sin\left(\frac{\nu x}{l}\right)\right] \frac{l^3}{\nu^3 EI} & \left[1 - \cos\left(\frac{\nu x}{l}\right)\right] \frac{l^2}{\nu^2 EI} \\ 0 & 0 & \cos\left(\frac{\nu x}{l}\right) & 0 & \left[\cos\left(\frac{\nu x}{l}\right) - 1\right] \frac{l^2}{\nu^2 EI} & \left[-\sin\left(\frac{\nu x}{l}\right)\right] \frac{l}{\nu EI} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -\nu \sin\left(\frac{\nu x}{l}\right) \frac{EI}{l} & 0 & -\sin\left(\frac{\nu x}{l}\right) \frac{l}{\nu} & -\cos\left(\frac{\nu x}{l}\right) \end{bmatrix} \quad (5.110)$$

and $\mathbf{U}_x^{\mathbf{S}+2} =$

$$= \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -\text{sh}\left(\frac{\nu x}{l}\right) \frac{l}{\nu} & 0 & \left[\text{sh}\left(\frac{\nu x}{l}\right) - \frac{\nu x}{l}\right] \frac{l^3}{\nu^3 EI} & \left[\text{ch}\left(\frac{\nu x}{l}\right) - 1\right] \frac{l^2}{\nu^2 EI} \\ 0 & 0 & \text{ch}\left(\frac{\nu x}{l}\right) & 0 & \left[1 - \text{ch}\left(\frac{\nu x}{l}\right)\right] \frac{l^2}{\nu^2 EI} & \left[-\text{sh}\left(\frac{\nu x}{l}\right)\right] \frac{l}{\nu EI} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & \nu \text{sh}\left(\frac{\nu x}{l}\right) \frac{EI}{l} & 0 & -\text{sh}\left(\frac{\nu x}{l}\right) \frac{l}{\nu} & -\text{ch}\left(\frac{\nu x}{l}\right) \end{bmatrix} \quad (5.111)$$

The transfer matrices (for the axial and transverse shear forces S , H ; Sign Convention 2) $\mathbf{U}_x^{\mathbf{S}-2}$ of Eq. (5.110) and $\mathbf{U}_x^{\mathbf{S}+2}$ of Eq. (5.111) can be computed using the GNU Octave function `yfmhvII.m` (p. 248), taking the input argument `baasi0` equal to 1.0 (`baasi0` is input for scaling multiplier $i_0 = EI/l$, see p. 27).

Multiplying the loading vectors of Eqs. (5.92) and (5.93) (for the normal and shear forces N , Q) from left by $\mathbf{T}^{\mathbf{SN}}$ (see Eq. (5.109)), we get the products of Eqs. (5.112) and (5.113) (for the axial and transverse shear forces S , H):

$$\overset{\circ}{\mathbf{Z}}^{S-2} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ \left[\frac{1}{2} \left(\nu \frac{(x-x_0)}{l}\right)^2 - 1 + \cos\left(\nu \frac{(x-x_0)}{l}\right)\right] \frac{ql^4}{\nu^4 EI} \\ \left[\sin\left(\nu \frac{(x-x_0)}{l}\right) - \nu \frac{(x-x_0)}{l}\right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ -q \cdot (x - x_0) \\ \left[\cos\left(\nu \frac{(x-x_0)}{l}\right) - 1\right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (5.112)$$

$$\mathbf{Z}^{S+2} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ -\left[\frac{1}{2} \left(\nu \frac{(x-x_0)}{l} \right)^2 - 1 - \operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^4}{\nu^4 EI} \\ \left[\nu \frac{(x-x_0)}{l} - \operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ -q \cdot (x - x_0) \\ \left[1 - \operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (5.113)$$

The loading vectors of Eqs. (5.112) and (5.113) can be computed using the GNU Octave function `ylqvII.m` (p. 249), taking the input argument `baasi0` equal to 1.0 (`baasi0` is input for the scaling multiplier $i_0 = EI/l$, see p. 27).

Multiplying the loading vectors of Eqs. (5.94) and (5.95) (for the normal and shear forces N, Q) from left by \mathbf{T}^{SN} (see Eq. (5.109)), we get the products of Eqs. (5.114) and (5.115) (for the axial and transverse shear forces S, H):

$$\mathbf{Z}^{S-2} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ -\left[\sin \left(\nu \frac{(x-x_0)}{l} \right) - \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl^3}{\nu^3 EI} \\ \left[\cos \left(\nu \frac{(x-x_0)}{l} \right) - 1 \right] \frac{Fl^2}{\nu^2 EI} \\ 0 \\ -(x - x_0)_+^0 F \\ -\left[\sin \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl}{\nu} \end{bmatrix} \quad (5.114)$$

$$\mathbf{Z}^{S+2} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ \left[\operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) - \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl^3}{\nu^3 EI} \\ \left[1 - \operatorname{ch} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl^2}{\nu^2 EI} \\ 0 \\ -(x - x_0)_+^0 F \\ -\left[\operatorname{sh} \left(\nu \frac{(x-x_0)}{l} \right) \right] \frac{Fl}{\nu} \end{bmatrix} \quad (5.115)$$

The loading vectors of Eqs. (5.114) and (5.115) can be computed using the GNU Octave function `ylfhvzII.m` (p. 248), taking the input argument `baasi0` equal to 1.0 (`baasi0` is input for the scaling multiplier $i_0 = EI/l$, see p. 27).

6. The EST method for a second-order theory

6.1 System equations for a frame

Consider solving a boundary value problem with the EST method. The second-order analysis is a nonlinear problem and the superposition of deflections cannot be applied. Axial forces are generally not known at the outset of a frame analysis. A set of axial forces of the frame members is determined as a linear structure. At the second iteration, the axial forces from the first iteration are used. If the axial forces obtained by the second iteration differ greatly from the values of the first iteration, the calculated values are used to find new values and the analysis is repeated.

As can be seen from the computing diary excerpt 6.1, the axial forces of the frame elements (given in Fig. 6.7), obtained in the third iteration, do not differ from the values of the second iteration.

Computing diary excerpt 6.1

SIVec =

	Linear	1	2	3	4
1	-824.978	-828.823	-828.821	-828.821	-828.821
2	-25.760	-21.535	-21.499	-21.499	-21.499
3	-775.104	-773.182	-773.186	-773.186	-773.186
4	-79.661	-77.786	-77.769	-77.769	-77.769
5	-739.918	-737.994	-737.993	-737.993	-737.993

In excerpt 6.2 from the computing diary, an iteration for the support reactions of the frame from Fig. 6.7 is shown.

Computing diary excerpt 6.2

Support_Reactions =

	Linear	1	2	3	4
Cx	25.760	21.535	21.499	21.499	21.499
Cz	-824.978	-828.823	-828.821	-828.821	-828.821
Cx	53.901	56.251	56.270	56.270	56.270
Cz	-775.104	-773.182	-773.186	-773.186	-773.186
Cy(moment)	-138.774	-157.125	-157.134	-157.134	-157.134
Cx	130.339	132.214	132.231	132.231	132.231
Cz	-739.918	-737.994	-737.993	-737.993	-737.993
Cy(moment)	-220.868	-239.506	-239.508	-239.508	-239.508

Program excerpt 6.1 (The basic equations. `Lahe2FrameDFIm.m`)

```

IIv=0;
IJv=0;
#
for i=1:NEARV
    qkoormus=zeros(1,3);
    Fjoud=zeros(1,2);
    spvF=zeros(4,8);
    vB=zeros(4,1);
    vFz=zeros(4,1);
    krda=i;
    EI=selem(i,13);
    EA=selem(i,14);
    GAr=selem(i,15);
    Li=lvarras(i,1);
    SII=SI(i,1);
    qzZ(i,1)=esQkoormus(1,1,i);
    FZz(i,1)=esFjoud(1,1,i);
    aLXx(i,1)=esFjoud(1,3,i);
    qz=qzZ(i,1);
    aLx=aLXx(i,1);
    Fz=FZz(i,1);
#    Fx=FZx(i,1);
#    -----The transfer matrix equation -----
    spvF=ysplvmhvII(baasi0,SII,Li,Li,EA,GAr,EI);
    vB=ylqvII(baasi0,SII,Li,Li,qz,EA,EI);
    vFz=ylfhvzII(baasi0,SII,Li,Li,aLx,Fz,EI);
    vB=vB+vFz;
    IIv=krda*6-5;
    IJv=krda*12-11;
    spA=spInsertBtoA(spA,IIv,IJv,spvF);
    B=InsertBtoA(B,NNK,1,IIv,1,vB,6,1);
endfor
#
% here NEARV — the number of elements
% baasi0 — the scaling multiplier for displacements

```

Let us start by assembling a system of non-symmetric sparse equations

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (6.1)$$

The system (6.1) is a collection of the

1. basic equations of a frame in transfer matrix form,
2. compatibility equations of the displacements at nodes,
3. joint equilibrium equations,
4. side conditions (hinges),
5. restrictions on support displacements.

This collection of boundary problem equations (6.1) is assembled and solved by the GNU Octave function `Lahe2FrameDFIm.m` (p. 246).

The basic equations of a frame are defined as

$$\widehat{\mathbf{IU}}_{6 \times 12} \cdot \widehat{\mathbf{Z}}_{12 \times 1} = \widehat{\mathbf{Z}}_{6 \times 1} \quad (6.2)$$

where $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{Z}}$ are expressed as

$$\widehat{\mathbf{U}}_{6 \times 12} = \mathbf{I}_{6 \times 6} - \mathbf{U}_{6 \times 6}, \quad (6.3)$$

$$\widehat{\mathbf{Z}}_{12 \times 1} = \begin{bmatrix} \mathbf{Z}_L \\ \mathbf{Z}_A \end{bmatrix} \quad (6.4)$$

Here, $\mathbf{U}_{6 \times 6} \equiv \mathbf{U}_x^{S \mp 2}$ is the transfer matrix given with Eqs. (5.110) and (5.111) at $x = l$ (Sign Convention 2); \mathbf{Z}_L , \mathbf{Z}_A are the vectors of displacements and forces at the end and at the beginning of the element, respectively; the loading vector $\mathring{\mathbf{Z}} \equiv \mathring{\mathbf{Z}}^{S \mp 2}$ is given with Eqs. (5.112), (5.113), (5.114), and (5.115); $\mathbf{I}_{6 \times 6}$ is a unit matrix 6×6 for the frame element.

Equilibrium equations at beam joints and joint equilibrium equations are discussed in sections 2.2 and 2.3, side conditions and restrictions on support displacements are dealt with in sections 2.4 and 2.5.

Inserting the system of the basic equations (6.2) into the system Eq. (6.1) is shown in excerpt 6.1 of the program.

6.2 Illustrative frame problem 1

Example 6.1. Problem Statement. Figure 6.1 shows a frame of the span length $l = 6$ m, carrying a uniform load $q = 20$ kN/m. The 5 m long column 3–4 and the 2.5 m long column 2–3 are both loaded with a vertical force $F = 800$ kN. At node 2, a horizontal force $H = 90$ kN acts.

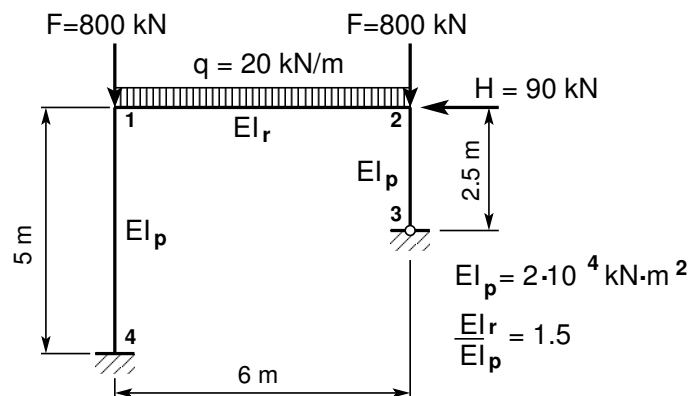


Figure 6.1. The frame EST1

We assume that the value of the column flexural rigidity $EI_p = 2 \cdot 10^4$ kN·m² and that of the beam $EI_r = 1.5EI_p$ ($I_1 = 1.5 I_2$); axial rigidity of the columns $EAp = 2.0 \cdot 10^{20}$ kN and that of the beam $EAr = 2.0 \cdot 10^{20}$ kN; shear rigidity of the columns $GAp = 1.0 \cdot 10^{25}$ and that of the beam $GAr = 1.0 \cdot 10^{25}$.

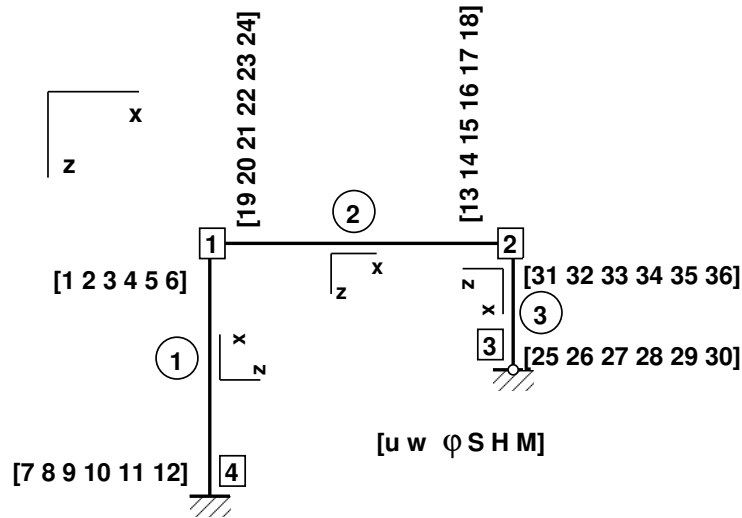


Figure 6.2. Numeration of displacements and forces of the frame EST1

We wish to compute the displacements, reactions, internal forces, and draw the axial force, shear force and bending moment diagrams.

Problem Solving. To solve the problem, we use the EST method. The solving procedure includes the following.

1. *Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, restrictions on support displacements.*

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (6.5)$$

2. *Assembling and solving the boundary problem equations (6.1) and (6.5) (prepared and solved by the program). To implement this aim, the program*

- (a) *inserts the basic equations of a frame into the equation system,*
- (b) *adds the compatibility equations of the displacements at nodes to the equation system,*
- (c) *adds the joint equilibrium equations,*
- (d) *adds the side conditions (hinges),*
- (e) *adds the restrictions on support displacements,*
- (f) *solves the system of sparse equations,*
- (g) *produces an output: initial parameter vectors for element displacements and forces; support reactions.*

3. *Output: element displacements and forces determined by the transfer matrix.*

1. **Input data** for the GNU Octave program `yspESTframe1LaheWFI.m` are given in excerpts from the program: element and nodal loads – excerpt 6.2; nodal coordinates – excerpt 6.3; element properties, topology and hinges – excerpt 6.4.

Program excerpt 6.2 ("Nmitmeks" yspESTframe1LaheWFI.m)

```

epsdountil=0.0000001; # for iteration (max |S| increment)
Number_of_frame_nodes=4
Number_of_elements=3
Number_of_support_reactions=5
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts ''Nmitmeks'' of the element
Nmitmeks=4
# --- Element properties ---

EIp=20000; # kN/m^2                                # EAr=6.8*10^6;
EIr=40000; # kN/m^2                                #   Gap=1.0e+25;
# EAp=4.6*10^6;                                     #   Gar=1.0e+25;
EAp=2.0^20;                                         #   Gap=0.4*EAp;
# EAp=4.6*10^6;                                     #   Gar=0.4*EAr;
EAr=6.8*10^20;

baasi0=EIp/5 # scaling multiplier for displacements
# baasi0=1.0;
# Element load in local coordinates
# qz   qx   qA   qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

baasi0=EIp/5 # scaling multiplier for the displacements
esQkoormus(1,1:4,1)=[0.0 0.0 0.0 5.0];   esQkoormus(1,1:4,2)=[20.0 0.0 0.0 6.0];
esQkoormus(1,1:4,3)=[0.0 0.0 0.0 2.5];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 5.0];           esFjoud(1,1:3,3)=[0.0 0.0 2.5];
esFjoud(1,1:3,2)=[0.0 0.0 6.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0                               sSolmF(2,1,2)= 800.0; % Fz
sSolmF(2,1,1)= 800.0; % Fz                         #sSolmF(:,1,3)= 0.0
sSolmF(1,1,2)= -90.0; % Fx                          #sSolmF(:,1,4)= 0.0

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

```

```
#tSiire(:,1,1)= 0.0                #tSiire(:,1,3)= 0.0
#tSiire(2,1,1)= 0.01*baasi0        #tSiire(:,1,4)= 0.0
#tSiire(:,1,2)= 0.0
```

Program excerpt 6.3 (`ySpESTframe1LaheWFI.m`)

```
#####
#      Nodal coordinates
#####
krdn=[#  x      z
      0.0    0.0;   % node 1
      6.0    0.0;   % node 2
      6.0    2.5;   % node 3
      0.0    5.0]; % node 4
#####

#####
# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#####
tsolm=[3      1 1 0;   % node 3
      4      1 1 1]; % node 4
#####
```

Program excerpt 6.4 (`ySpESTframe1LaheWFI.m`)

```
# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#          n2 - end of the element
#          n1 - beginning of the element
#          N, Q, M - hinges at the end of the element
#          N, Q, M - hinges at the beginning of the element
#
EIp EAp GAp  1    4    0 0 0    0 0 0;   % element 1
EIr EAr GAR  2    1    0 0 0    0 0 0;   % element 2
EIp EAp GAp  3    2    0 0 1    0 0 0]; % element 3
# 1 - hinge 'true' (axial, shear, moment hinges)
#
```

2. *Assembling and solving the boundary problem equations (6.5), carried out by the function `Lahe2FrameDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the element ends of the frame as shown in Fig. 6.2.*

The results of iteration of the axial forces are shown in excerpt 6.3 from the computing diary.

Computing diary excerpt 6.3 (`ySpESTframe1LaheWFI.m`)

```
Sivec =
Element  Linear      1          2          3          4          5
1      -897.500 -904.385 -904.364 -904.364 -904.364 -904.364
2       -54.000 -55.431 -55.362 -55.362 -55.362 -55.362
3      -822.500 -815.615 -815.636 -815.636 -815.636 -815.636
```

The unscaled initial parameter vectors of the elements are shown in excerpt 6.4 from the computing diary.

Computing diary excerpt 6.4 (yspESTframe1LaheWFI.m)

----- Scaling multiplier for displacements = 1/baasi0 -----

=====

```

Unscaled initial parameter vector
Element No   u           w           fi           S           H           M
-----
  1  -0.000e+00  0.000e+00  0.000e+00   904.364    55.362   -154.733
  2  -3.344e-02  2.261e-17  3.342e-04    55.362   -104.364   152.316
  3   1.020e-17  3.344e-02  8.548e-03   815.636    34.638   -113.870
-----

```

The support reactions of the frame in global coordinates are shown in excerpt 6.5 from the computing diary.

Computing diary excerpt 6.5 (yspESTframe1LaheWFI.m)

Support reactions begin from X row: 37

```

=====
No           X           Node    Cx <=> 1
              Cz <=> 2
              Cy <=> 3
-----
37   +3.463836e+01    3         1
38   -8.156356e+02    3         2
39   +5.536164e+01    4         1
40   -9.043644e+02    4         2
41   -1.547333e+02    4         3
-----

```

3. Output: the element displacements and forces determined by the transfer matrix are given in excerpt 6.6 from the computing diary.

The bending moment, shear force Q and axial force N diagrams of the frame EST1 are shown in Fig. 6.3.

Computing diary excerpt 6.6 (yspESTframe1LaheWFI.m)

```

#=====
Element displacements and forces determined by transfer matrix
#=====
Displacements and forces of element no 1 of length 5.000 m
The element is divided into 4 parts
displacement u - 0.00000e+00 -5.65228e-18 -1.13046e-17 -1.69568e-17 -2.26091e-17
displacement w - 0.00000e+00 -5.11088e-03 -1.65057e-02 -2.80093e-02 -3.34392e-02
rotation fi - 0.00000e+00 7.40749e-03 9.99497e-03 7.58070e-03 3.34242e-04

normal force N - -904.36438 -904.36438 -904.36438 -904.36438 -904.36438
shear force Q - -55.36164 -62.06071 -64.40074 -62.21735 -55.66392
moment force M - 154.73331 80.90916 1.40208 -78.20348 -152.31615
-----
axial force S - -904.36438 -904.36438 -904.36438 -904.36438 -904.36438
transv force H - -55.36164 -55.36164 -55.36164 -55.36164 -55.36164
-----
Displacements and forces of element no 2 of length 6.000 m

```

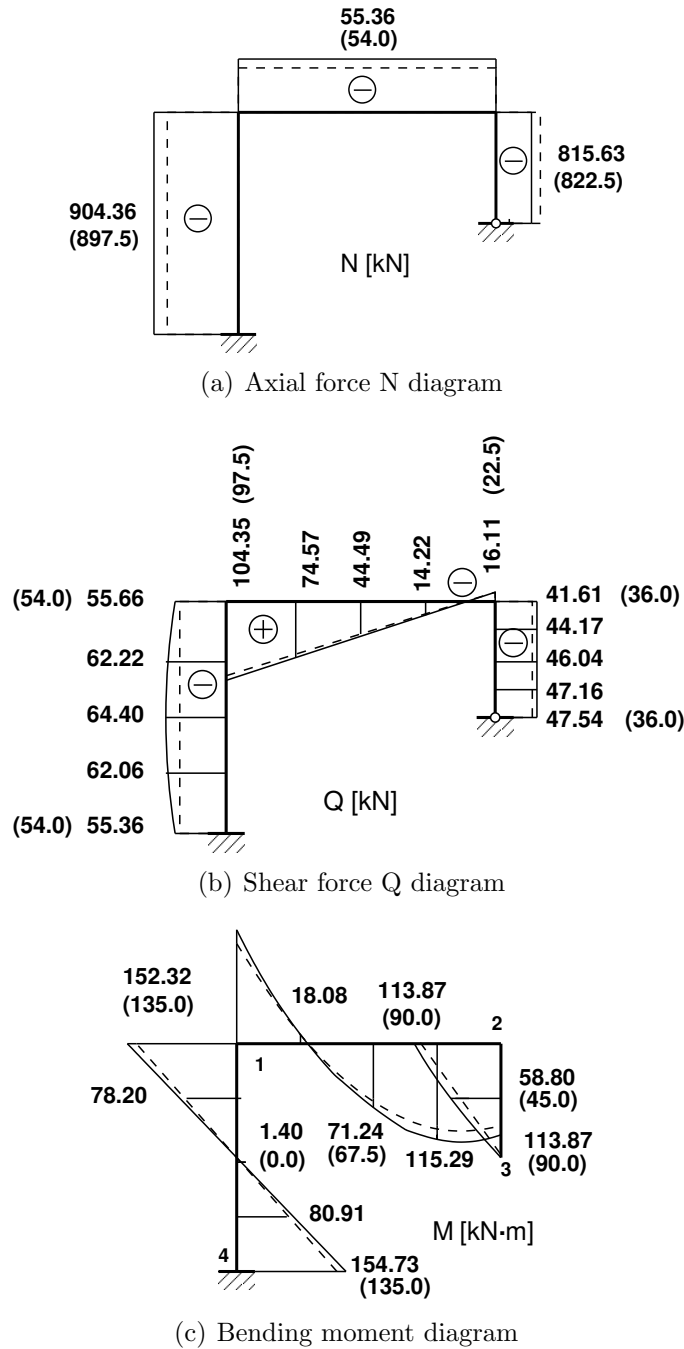


Figure 6.3. Internal forces diagrams of the frame EST1
(bracketed are values of linear solution)

The element is divided into 4 parts

displacement u	-3.34392e-02	-3.34392e-02	-3.34392e-02	-3.34392e-02	-3.34392e-02
displacement w	2.26091e-17	3.39304e-03	8.42296e-03	8.39268e-03	3.29397e-10
rotation fi	3.34242e-04	-3.73960e-03	-2.22252e-03	2.62995e-03	8.54847e-03
normal force N	-55.36164	-55.36164	-55.36164	-55.36164	-55.36164
shear force Q	104.34587	74.57141	44.48742	14.21878	-16.10888
moment force M	-152.31615	-18.08174	71.24329	115.28819	113.87012

axial force S	-	-55.36164	-55.36164	-55.36164	-55.36164	-55.36164
transv force H	-	104.36438	74.36438	44.36438	14.36438	-15.63562

Displacements and forces of element no 3 of length 2.500 m
The element is divided into 4 parts

displacement u	-	1.01954e-17	7.64658e-18	5.09772e-18	2.54886e-18	1.54074e-33
displacement w	-	3.34392e-02	2.70705e-02	1.90043e-02	9.79115e-03	7.61059e-11
rotation fi	-	8.54847e-03	1.16917e-02	1.39732e-02	1.53568e-02	1.58205e-02
normal force N	-	-815.63562	-815.63562	-815.63562	-815.63562	-815.63562
shear force Q	-	-41.61080	-44.17449	-46.03540	-47.16392	-47.54210
moment force M	-	113.87012	87.02657	58.79850	29.63498	-0.00000

axial force S	-	-815.63562	-815.63562	-815.63562	-815.63562	-815.63562
transv force H	-	-34.63836	-34.63836	-34.63836	-34.63836	-34.63836

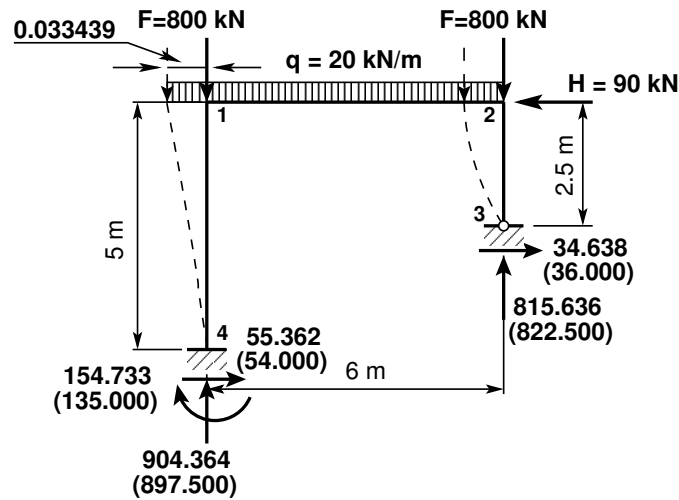


Figure 6.4. Boundary forces of the frame EST1

Testing a static equilibrium for the frame

Consider next a static equilibrium of the frame shown in Fig. 6.4. Let us project the forces onto the X-axis,

$$\Sigma X = 0; \quad 55.36164 + 34.63836 - 90 = 0.0 \quad (6.6)$$

and onto the Z-axis,

$$\Sigma Z = 0; \quad -904.3644 - 815.6356 + 6 \cdot 20 + 2 \cdot 800 = 0.0 \quad (6.7)$$

We now write the equation of the sum of the moments and the moments of the forces acting about point 3 shown in Fig. 6.4:

$$\begin{aligned} \Sigma M_3 = 0; \quad & -154.7333 - 904.3644 \cdot 6.0 + 55.36164 \cdot 2.5 + 800 \cdot 0.0334392 \\ & + 800 \cdot (6.0 + 0.0334392) + 20 \cdot 6.0 \cdot (3 + 0.0334392) + 90 \cdot 2.5 = -1.76 \cdot 10^{-4} \text{ kN}\cdot\text{m} \end{aligned} \quad (6.8)$$

The calculations with equations (6.6), (6.7), and (6.8) have verified the static equilibrium of the frame.

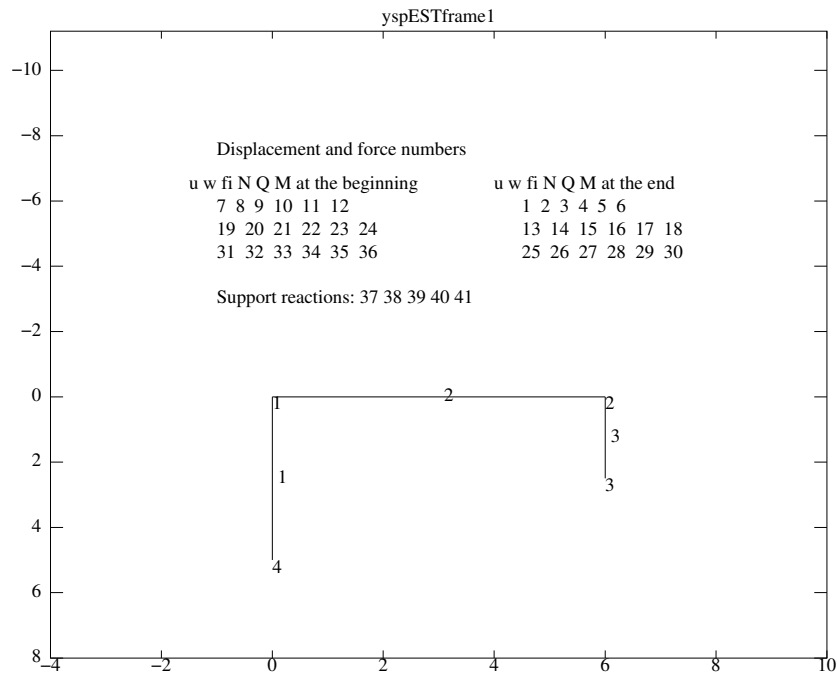


Figure 6.5. Elements of the frame EST1

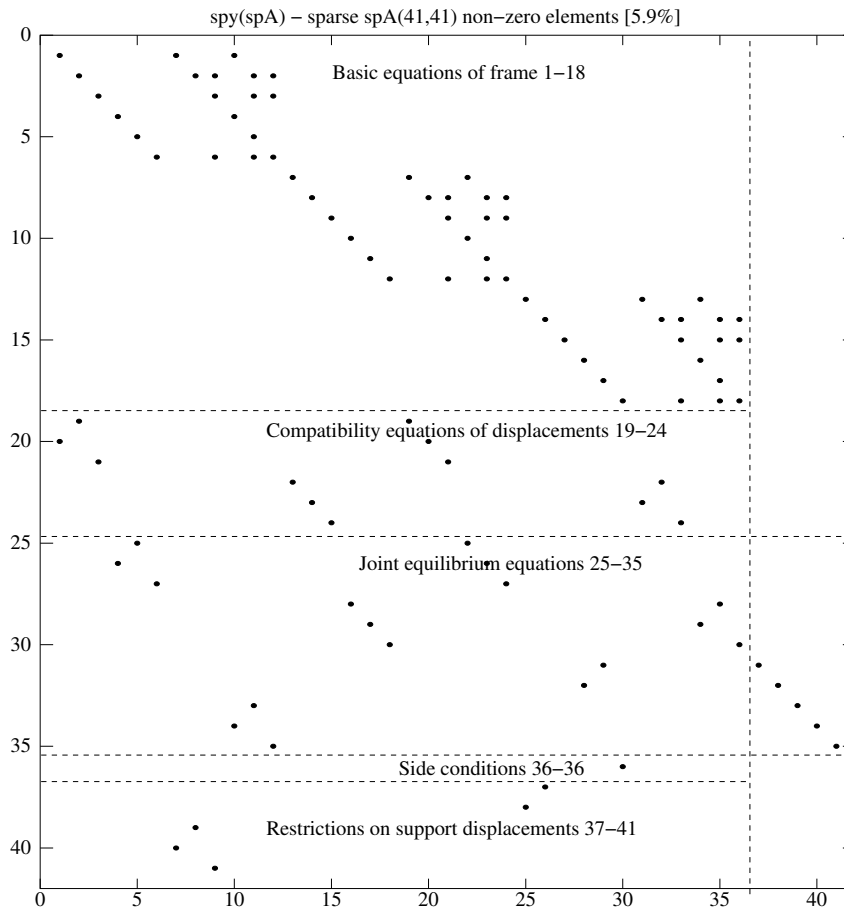


Figure 6.6. Sparsity pattern of matrix spA of the frame EST1

6.3 Illustrative frame problem 2

Example 6.2. Problem Statement. In Fig. 6.7, a two-span frame of the height $h = 4$ m is shown. Two spans of the frame are of the same length, $l = 6$ m. The beam of the first span carries a uniform load $q = 15$ kN/m. The rightmost column is loaded with horizontal concentrated loads $F_1 = 90$ kN and $F_2 = 120$ kN. Each of the frame columns carries a vertical concentrated load $F_3 = 750$ kN.

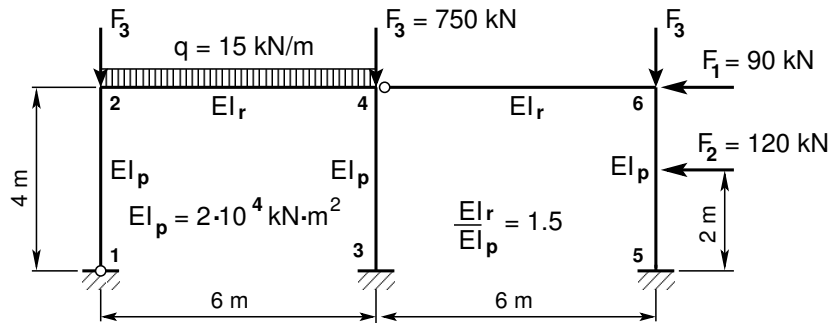


Figure 6.7. The two-span frame EST2

Let us assume that the value of the column flexural rigidity $EI_p = 2 \cdot 10^4$ kN·m² and beam flexural rigidity $EI_r = 1.5EI_p$; axial rigidity of the columns $EA_p = 4.6 \cdot 10^6$ kN and that of the beam $EA_r = 8.8 \cdot 10^6$ kN; shear rigidity of the columns $GA_p = 0.4EA_p$ and that of the beam $GA_r = 0.4EA_r$.

We wish to compute the displacements, reactions, internal forces, and draw the axial force, shear force and bending moment diagrams.

Problem Solving. We use the EST method to solve the problem. The solving procedure includes the following.

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, restrictions on support displacements.
2. Assembling and solving the boundary problem equations (6.1) and (6.9) (prepared and solved by the program):

(a) writing the basic equations of a frame in transfer matrix form,

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (6.9)$$

(b) adding the compatibility equations of the displacements at nodes,

(c) adding the joint equilibrium equations,

(d) adding the side conditions (hinges),

(e) adding the restrictions on support displacements,

(f) solving the compiled system of sparse equations,

(g) producing an output: initial parameter vectors for element displacements and forces; support reactions.

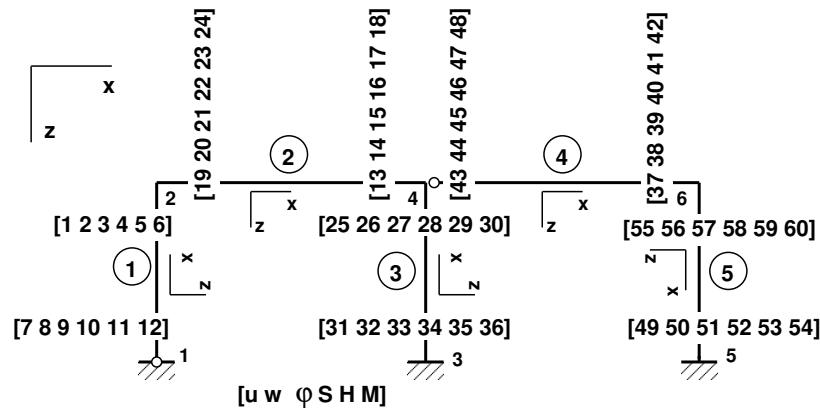


Figure 6.8. Displacement and force numbers of the two-span frame EST2

3. *Output: element displacements and forces determined by the transfer matrix.*

1. **Input data** for the GNU Octave program *yspESTframe2LaheWFI.m* are shown in excerpt from the program: element and nodal loads – excerpt 6.5; nodal coordinates – excerpt 6.6; element properties, topology and hinges – excerpt 6.7.

Program excerpt 6.5 (*yspESTframe2LaheWFI.m*)

```

epsdowntil=0.0000001; # for iteration (max |S| increment)
Number_of_frame_nodes=6
Number_of_elements=5
Number_of_support_reactions=8
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts ''Nmitmeks'' of the element
Nmitmeks=4
Lp=12.0; # graphics axis
# --- Element properties ---
    EIp=20000 # kN/m^2
    EIr=40000 # kN/m^2
    EAp=4.6*10^6
    #EAp=4.6*10^15;
    EAr=6.8*10^6;
    #EAr=6.8*10^15
    GAp=0.4*EAp;
    GAR=0.4*EAr;

baasi0=EIp/5.6 # scaling multiplier for the displacements
#baasi0=1.0;

#Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq0Nelement=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[15.0 0.0 0.0 6.0];
esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,4)=[0.0 0.0 0.0 6.0];
esQkoormus(1,1:4,5)=[0.0 0.0 0.0 4.0];

```

```

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF_on_Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];          esFjoud(1,1:3,4)=[0.0 0.0 6.0];
esFjoud(1,1:3,2)=[0.0 0.0 6.0];          esFjoud(1,1:3,5)=[120.0 0.0 2.0];
esFjoud(1,1:3,3)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0                      #sSolmF(:,1,5)= 0.0
sSolmF(2,1,2)= 750.0; % Fz                sSolmF(1,1,6)= -90.0; % Fx
#sSolmF(:,1,3)= 0.0                      sSolmF(2,1,6)= 750.0; % Fz
sSolmF(2,1,4)= 750.0; % Fz

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0                      #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0              #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0                      #tSiire(:,1,6)= 0.0
#tSiire(:,1,3)= 0.0

Program excerpt 6.6 ( yspESTframe2LaheWFI.m )
#=====
# Nodal coordinates
#=====
krdn=[# x z
      0.0 0.0; % node 1
      0.0 -4.0; % node 2
      6.0 0.0; % node 3
      6.0 -4.0; % node 4
      12.0 0.0; % node 5
      12.0 -4.0]; %% node 6
#=====
#
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No u w fi
#=====
tsolm=[1 1 1 0; % node 1

```

```

      3      1 1 1;  % node 3
      5      1 1 1]; % node 5
#=====

```

Program excerpt 6.7 (`yspESTframe2LaheWFI.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#         n2 - end of the element
#         n1 - beginning of the element
#         N, Q, M - hinges at the end of the element
#         N, Q, M - hinges at the beginning of the element
#
EIp EAp GAp  2      1      0 0 0      0 0 1;  % element 1
EIr EAr GAR  4      2      0 0 0      0 0 0;  % element 2
EIp EAp GAp  4      3      0 0 0      0 0 0;  % element 3
EIr EAr GAR  6      4      0 0 0      0 0 1;  % element 4
EIp EAp GAp  5      6      0 0 0      0 0 0]; % element 5
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (6.5), carried out by the function `Lahe2FrameDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. The program has numbered the displacements and forces of the element ends of the frames as shown in Figs. 6.2 and 6.8.*

The results of iteration of the axial forces are given in excerpt 6.7 from the computing diary.

Computing diary excerpt 6.7 (`yspESTframe2LaheWFI.m`) S1vec =

Element	Linear	1	2	3	4
1	-824.978	-828.823	-828.821	-828.821	-828.821
2	-25.760	-21.535	-21.499	-21.499	-21.499
3	-775.104	-773.182	-773.186	-773.186	-773.186
4	-79.661	-77.786	-77.769	-77.769	-77.769
5	-739.918	-737.994	-737.993	-737.993	-737.993

The unscaled initial parameter vectors of the elements are shown in excerpt 6.8 from the computing diary.

Computing diary excerpt 6.8 (`yspESTframe2LaheWFI.m`)

----- Scaling multiplier for displacements = 1/baasi0 -----

```

=====
Unscaled initial parameter vector
Element No   u           w           fi           S           H           M
-----
  1  -0.000e+00  0.000e+00  1.166e-02   828.821    21.499     0.000
  2  -3.056e-02  7.207e-04 -1.311e-04   21.499   -78.821    111.330
  3  -0.000e+00  0.000e+00  0.000e+00   773.186    56.270   -157.134
  4  -3.058e-02  6.723e-04 -2.423e-03    77.769   -12.007     0.000
  5   6.417e-04  3.065e-02  4.838e-03   737.993    12.231   -72.037
-----

```

The support reactions of the frame in global coordinates are shown in excerpt 6.9 from the computing diary.

Computing diary excerpt 6.9 (yspESTframe2LaheWFI.m)

Support reactions begin from X row: 61

```
=====
No      X      Node  Cx <=> 1
                Cz <=> 2
                Cy <=> 3
-----
61  +2.149932e+01   1      1
62  -8.288206e+02   1      2
63  +5.626986e+01   3      1
64  -7.731859e+02   3      2
65  -1.571336e+02   3      3
66  +1.322308e+02   5      1
67  -7.379935e+02   5      2
68  -2.395078e+02   5      3
-----
```

3. **Output:** the element displacements and forces determined by the transfer matrix are given in excerpt 6.10 from the computing diary.

The horizontal displacements diagram of the frame EST2 is depicted in Fig. 6.9. The bending moment, shear force Q and axial force N diagrams of the frame EST2 are shown in Fig. 6.10.

Computing diary excerpt 6.10 (yspESTframe2LaheWFI.m)

```
=====
Element displacements and forces determined by transfer matrix
#=====
Displacements and forces of element no 1 of length 4.000 m
The element is divided into 4 parts
displacement u - 0.00000e+00 -1.80178e-04 -3.60357e-04 -5.40535e-04 -7.20714e-04
displacement w - 0.00000e+00 -1.14016e-02 -2.12611e-02 -2.81000e-02 -3.05647e-02
rotation fi - 1.16608e-02 1.08844e-02 8.58719e-03 4.86412e-03 -1.31089e-04
```

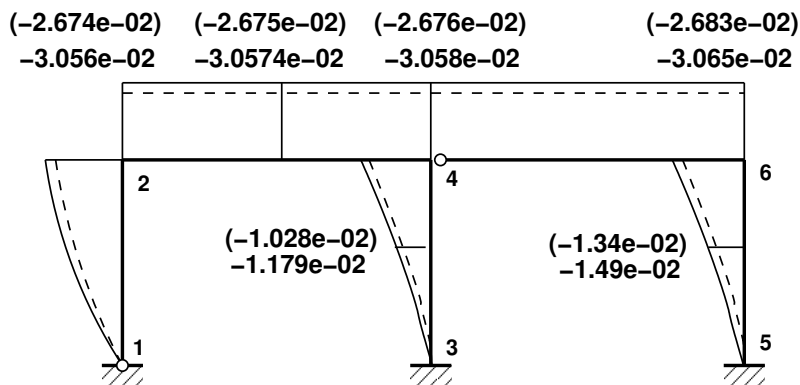
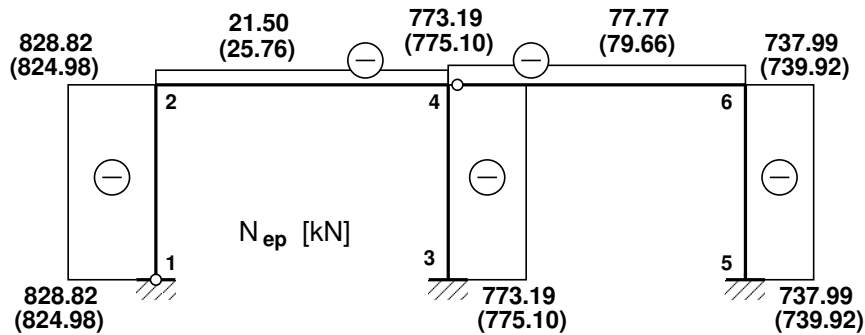
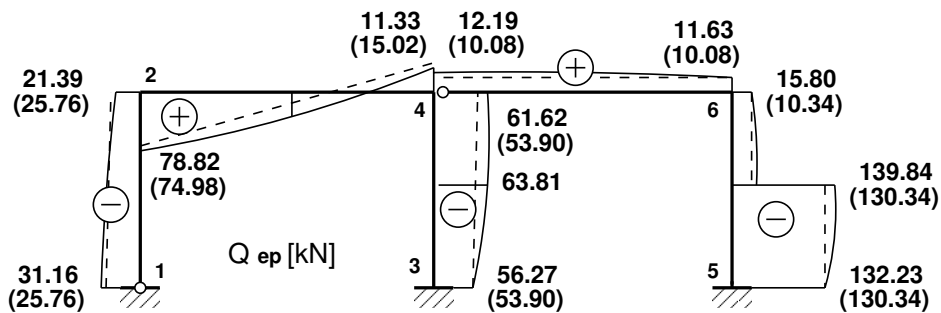


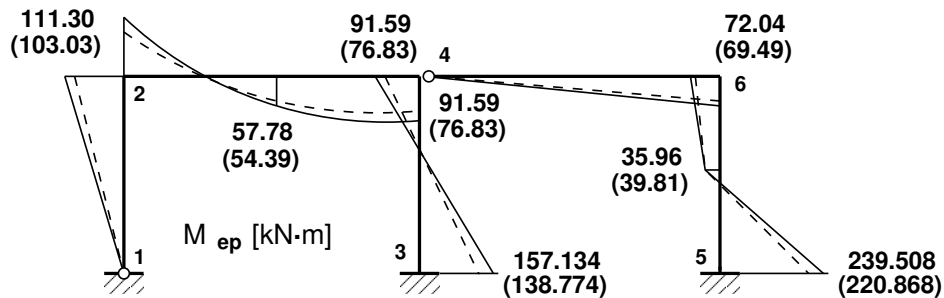
Figure 6.9. Horizontal displacements diagram of the frame EST2 (bracketed are values of linear solution)



(a) Axial force N diagram



(b) Shear force Q diagram



(c) Bending moment diagram

Figure 6.10. Internal forces diagrams of the two-span frame EST2
(bracketed are values of linear solution)

normal force N	-	-828.82062	-828.82062	-828.82062	-828.82062	-828.82062
shear force Q	-	-31.16401	-30.52051	-28.61656	-25.53080	-21.39067
moment force M	-	0.00000	-30.94922	-60.62029	-87.78785	-111.32995

axial force S	-	-828.82062	-828.82062	-828.82062	-828.82062	-828.82062
transv force H	-	-21.49932	-21.49932	-21.49932	-21.49932	-21.49932

Displacements and forces of element no 2 of length 6.000 m

The element is divided into 4 parts

displacement u	-	-3.05647e-02	-3.05695e-02	-3.05742e-02	-3.05789e-02	-3.05837e-02
displacement w	-	7.20714e-04	3.71930e-03	7.67193e-03	7.50247e-03	6.72336e-04
rotation fi	-	-1.31089e-04	-3.02184e-03	-1.68423e-03	2.19229e-03	6.91421e-03


```

normal force N - -21.49932  -21.49932  -21.49932  -21.49932  -21.49932
shear force Q -  78.82344   56.38559   33.85683   11.27349  -11.32803
moment force M - -111.32995  -9.90955   57.78135   91.63364   91.59272

```

```

-----
axial force S - -21.49932  -21.49932  -21.49932  -21.49932  -21.49932
transv force H -  78.82062   56.32062   33.82062   11.32062  -11.17938

```

Displacements and forces of element no 3 of length 4.000 m

The element is divided into 4 parts

```

displacement u - 0.00000e+00 -1.68084e-04 -3.36168e-04 -5.04252e-04 -6.72336e-04
displacement w - 0.00000e+00 -3.44769e-03 -1.17895e-02 -2.18995e-02 -3.05837e-02
rotation fi - 0.00000e+00 6.40393e-03 9.75665e-03 9.92896e-03 6.91421e-03
normal force N - -773.18589 -773.18589 -773.18589 -773.18589 -773.18589
shear force Q - -56.26986 -61.22129 -63.81357 -63.94679 -61.61583
moment force M - 157.13359 98.19802 35.47841 -28.60837 -91.59272

```

```

-----
axial force S - -773.18589 -773.18589 -773.18589 -773.18589 -773.18589
transv force H - -56.26986 -56.26986 -56.26986 -56.26986 -56.26986

```

Displacements and forces of element no 4 of length 6.000 m

The element is divided into 4 parts

```

displacement u - -3.05837e-02 -3.06008e-02 -3.06180e-02 -3.06351e-02 -3.06523e-02
displacement w - 6.72336e-04 4.07754e-03 6.11281e-03 5.41621e-03 6.41733e-04
rotation fi - -2.42253e-03 -1.96544e-03 -5.96843e-04 1.67528e-03 4.83769e-03
normal force N - -77.76918 -77.76918 -77.76918 -77.76918 -77.76918
shear force Q - 12.19491 12.15936 12.05292 11.87622 11.63028
moment force M - 0.00000 18.27458 36.44263 54.39821 72.03667

```

```

-----
axial force S - -77.76918 -77.76918 -77.76918 -77.76918 -77.76918
transv force H - 12.00651 12.00651 12.00651 12.00651 12.00651

```

Displacements and forces of element no 5 of length 4.000 m

The element is divided into 4 parts

```

displacement u - 6.41733e-04 4.81300e-04 3.20867e-04 1.60433e-04 0.00000e+00
displacement w - 3.06523e-02 2.41507e-02 1.49071e-02 4.86941e-03 -6.63532e-15
rotation fi - 4.83769e-03 8.02360e-03 1.03047e-02 8.60626e-03 5.15213e-15
normal force N - -737.99349 -737.99349 -737.99349 -737.99349 -737.99349
shear force Q - -15.80100 -18.15219 -139.83562 -138.58218 -132.23082
moment force M - 72.03667 55.00768 35.95515 -103.68339 -239.50781

```

```

-----
axial force S - -737.99349 -737.99349 -737.99349 -737.99349 -737.99349
transv force H - -12.23082 -12.23082 -132.23082 -132.23082 -132.23082

```

Testing a static equilibrium for the frame

Consider next a static equilibrium of the frame shown in Fig. 6.11. Let us project the forces onto the X-axis (see Fig. 6.11),

$$\Sigma X = 0; \quad 21.49932 + 56.26986 + 132.2308 - 90 - 120 = -2.0 \cdot 10^{-5} \quad (6.10)$$

and onto the Z-axis,

$$\Sigma Z = 0; \quad -828.8206 - 773.1859 - 737.9935 + 15 \cdot 6.0 + 3 \cdot 750 = 0.0 \quad (6.11)$$

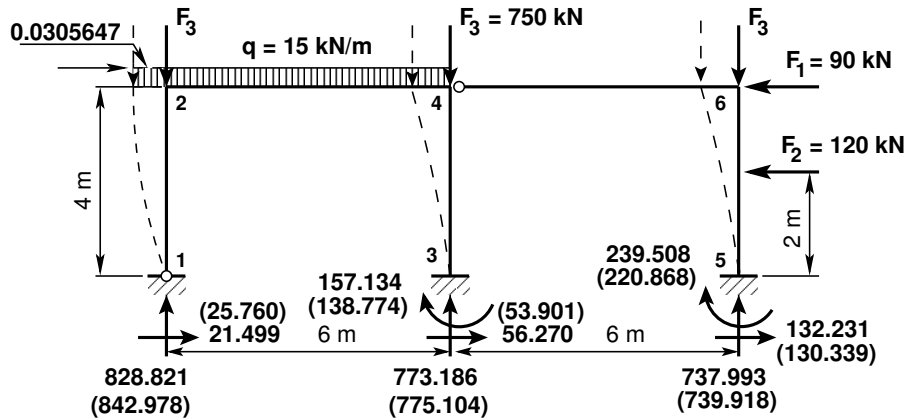


Figure 6.11. Boundary forces of the two-span frame EST2

We now write equation (6.12) of the sum of the moments and the moments of the forces acting about point 1 shown in Fig. 6.11:

$$\begin{aligned} \Sigma M_1 = 0; \quad & -157.1336 - 239.5078 + 773.1859 \cdot 6 + 737.9935 \cdot 12 + 120.0 \cdot 2.0 \\ & + 90.0 \cdot 4.0 - 750.0 \cdot (12.0 - 0.0306523) - 750.0 \cdot (6.0 - 0.0305837) \\ & - 15 \cdot 6.0 \cdot (3.0 - 0.0305742) + 0.0305647 \cdot 750.0 = -1.797 \cdot 10^{-3} \end{aligned} \quad (6.12)$$

At node 2, the horizontal displacement $u_{24} = -3.05647 \cdot 10^{-2}$ (see excerpt 6.10 from the computing diary). The additional moment ΔM from node (member) displacements

$$\begin{aligned} \Delta M = 0; \quad & -750.0 \cdot (-0.0306523) - 750.0 \cdot (-0.0305837) \\ & - 15 \cdot 6.0 \cdot (-0.0305742) + 0.0305647 \cdot 750.0 = 71.602 \text{ kN}\cdot\text{m} \end{aligned} \quad (6.13)$$

The calculations with equations (6.10), (6.11), and (6.12) have verified the static equilibrium of the frame.

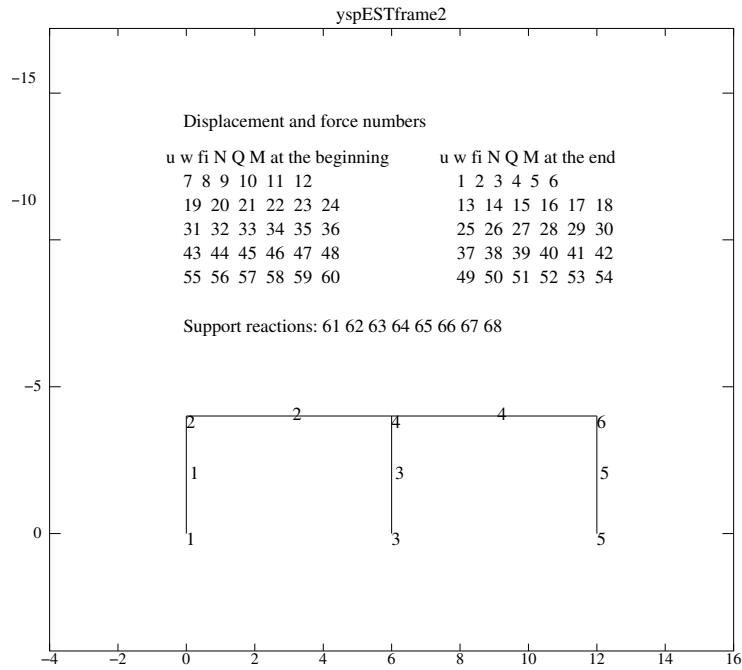


Figure 6.12. Elements of the frame EST2

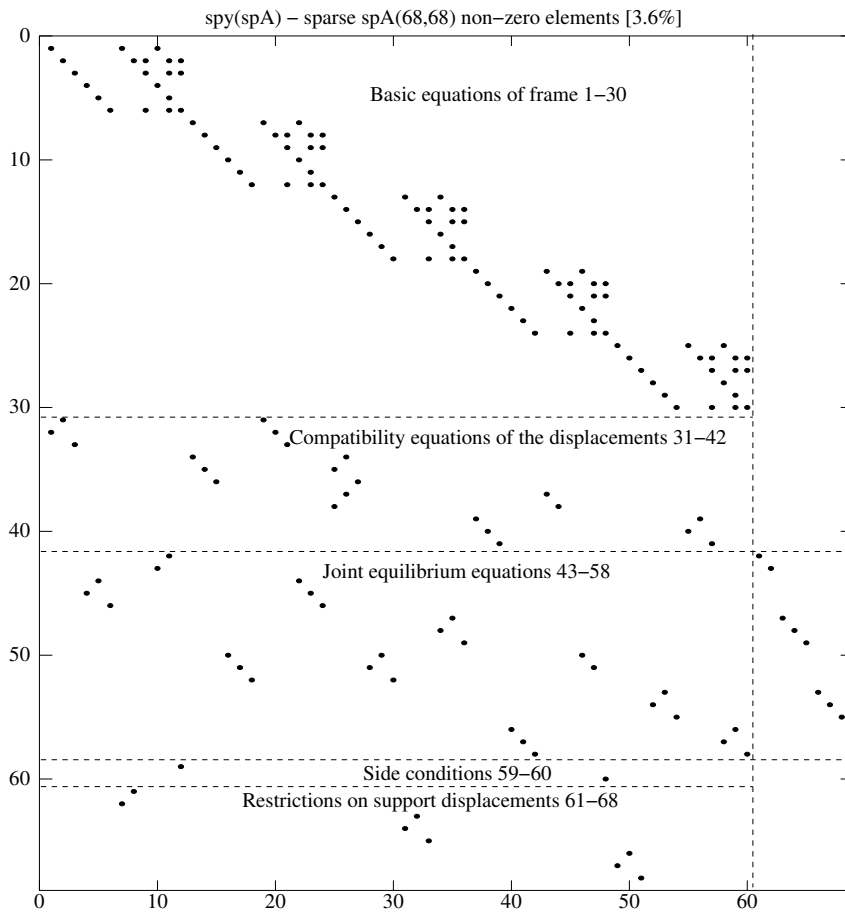


Figure 6.13. Sparsity pattern of matrix spA of the frame EST2

Table 6.1 gives the values for internal reactions^{1 2 3} (contact forces⁴) determined by the EST method. These can be compared with the results (M_{def})⁵ obtained with the slope-deflection method. The table describes the boundary values (internal reactions) after the second iteration, whereas those in table 16.17 of [Lah12] had been obtained in the first iteration.

Table 6.1. Comparison of the internal reactions values of the frame EST2

Element No	Linear theory				2nd order theory		
	M_{def} [kN·m]	M_{est}^I [kN·m]	S_{est}^I [kN]	H_{est}^I [kN]	M_{est}^{II} [kN·m]	S_{est}^{II} [kN]	H_{est}^{II} [kN]
1 begin.	0.0000	0.000	824.978	25.760	0.000	828.820	21.499
1 end	-103.1977	-103.040	-824.978	-25.760	-111.330	-828.820	-21.499
2 begin.	103.1977	103.040	25.760	-74.978	111.330	21.499	-78.820
2 end	76.9767	76.829	-25.760	-15.022	91.593	-21.499	-11.179
3 begin.	-138.8954	-138.774	775.104	53.901	-157.133	773.186	56.270
3 end	-76.9767	-76.829	-775.104	-53.901	-91.593	-773.186	-56.270
4 begin.	0.0000	0.000	79.661	-10.082	0.000	77.769	-12.007
4 end	60.3488	60.490	-79.661	10.082	72.037	-77.769	12.001
5 begin.	-60.3488	-60.490	739.918	10.339	-72.037	737.993	12.231
5 end	-220.5814	-220.868	-739.918	-130.339	-239.508	-737.993	-132.231

The boundary value problem (differential equations together with a set of boundary conditions) is well posed with the EST method.

¹Fixed-end forces and moments at joints are called internal reactions http://www.gunt.de/static/s3117_1.php?p1=&p2=&pN=;;. Web. 20 August 2013.

²http://engr.bd.psu.edu/rxm61/213/Beams_overview.ppt. Web. 20 August 2013.

³Shear and Moment Diagrams for Frames: ... cut the frame into its component members and find internal reactions http://www.ce.memphis.edu/3121/notes/notes_04d.pdf. Web. 20 August 2013.

⁴A contact force is a force that acts at the points of contact between two objects [Rand07].

⁵http://digi.lib.ttu.ee/opik_eme/Ehitusmehaanika.pdf.

Part III

Plastic analysis of frames

7. Plasticity and limit design

7.1 Introduction

An analysis of a structure is an important means for estimating its serviceability under the loading for which the structure has been designed. If the load is increased until the structure reaches a full yield, plastic hinges will form and the structure will transform to a mechanism. Under further loading this mechanism would collapse. The number of plastic hinges required to convert a structure into a mechanism is one more than the degree of indeterminacy ($n+1$). A study of the collapse load magnitude is necessary to determine the load factor.

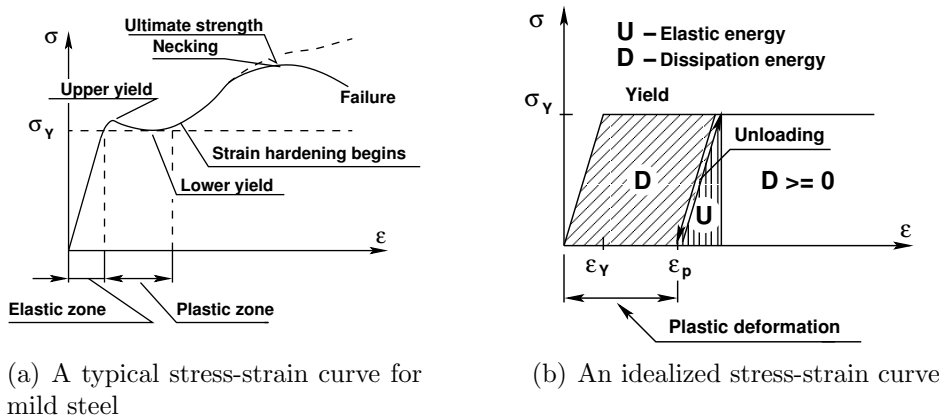


Figure 7.1. Stress-strain curve

The load factor λ_i [Cpr11] for a collapse mechanism i is

$$\lambda_i = \frac{\text{Collapse load for mechanism } i}{\text{Working load}} \quad (7.1)$$

where the collapse load is the load at which the structure will fail, and the working load is the load which the structure is expected to carry.

In *statically indeterminate structures* the collapse load is larger than the load that initiates yielding.

Let us consider an ideal elastic-plastic material yielding (see Prandtl¹ diagram Fig. 7.1 b) in the beam cross section.

¹Ludwig Prandtl (1875–1953), a German engineer.

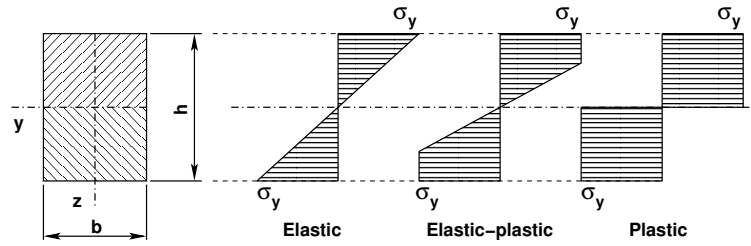


Figure 7.2. Rectangular cross section

In structural members subjected to bending, an assumption that a *plastic hinge* might form at a point is sufficient for limit state analysis. At a plastic hinge, stresses remain constant (see Fig. 7.2), whereas strains and rotations can increase. We assume that the condition for the axial force $N/N_Y \ll 1$ is fulfilled, otherwise plastic bending and compression or tension must be combined [IB01].

7.1.1 The plastic moment

For pure bending $N = 0$, we use Eq. (7.2). The plastic moment M_{ypl} about the neutral axis y in a plastic hinge can be computed from Eq. (7.3).

$$\int_{A_t} \sigma_Y dA - \int_{A_s} \sigma_Y dA = 0 \quad (7.2)$$

$$\int_{A_t} z \sigma_Y dA - \int_{A_s} z \sigma_Y dA = M_{ypl} \quad (7.3)$$

where σ_Y is the yield stress, A_t – an area in tension, A_s – an area in compression. Equations (7.2) and (7.3) give

$$A_t - A_s = 0, \quad (7.4)$$

$$|S_t| + |S_s| = \frac{M_{ypl}}{\sigma_Y} \quad (7.5)$$

where S_t and S_s are accordingly the first moment of the area in tension and that of the area in compression about neutral axis.

From Eq. (7.5) we get the expression for the plastic moment

$$M_{ypl} = W_{ypl} \sigma_Y \quad (7.6)$$

where the *plastic section modulus* M_{ypl} is given by

$$W_{ypl} = |S_t| + |S_s| \quad (7.7)$$

For a rectangular cross section shown in Fig. 7.2, the *plastic section modulus* is

$$W_{ypl} = S_t + S_s = \frac{1}{2}bh \left(\frac{h}{4} + \frac{h}{4} \right) = \frac{bh^2}{4} \quad (7.8)$$

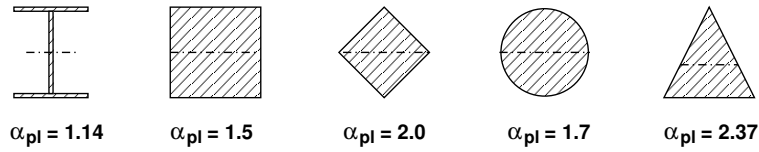


Figure 7.3. Shape factor α_{pl} for different sections

For a rectangular cross section, the *elastic section modulus* is

$$W_{yel} = \frac{bh^2}{6} \tag{7.9}$$

The ratio α_{pl} of the elastic section modulus to the plastic section modulus is

$$\alpha_{pl} = \frac{W_{ypl}}{W_{yel}} = \frac{\frac{bh^2}{4}}{\frac{bh^2}{6}} = 1.5 \tag{7.10}$$

This ratio α_{pl} is termed the *shape factor*. The shape factor is a measure of the efficiency of a cross section in bending. Shape factors for some cross sections are given in Fig. 7.3. Fig. 7.4 shows stresses at loading and unloading with a pure bending moment. At unloading, the stresses decrease (Fig. 7.4 g). The residual stresses on the cross section after removing a moment M_y are shown in Fig. 7.4 h.

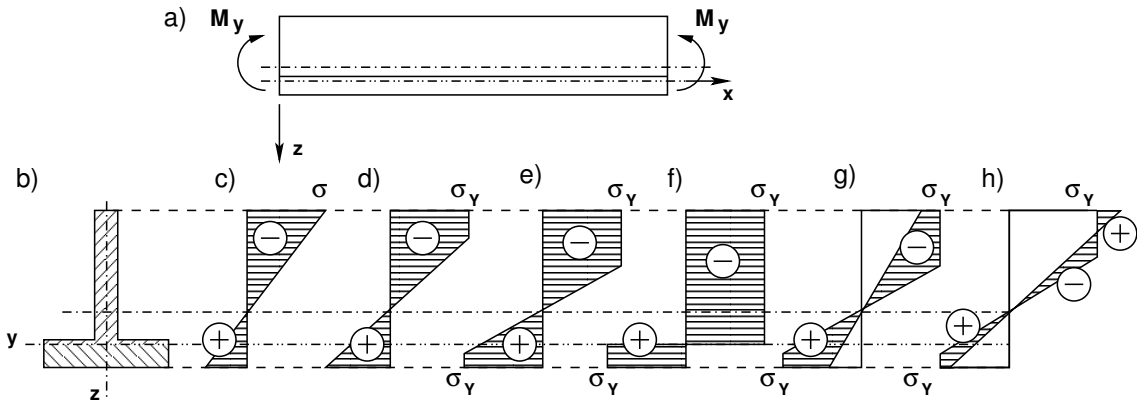


Figure 7.4. Cross section and stresses

7.1.2 Work done in a plastic hinge

Let us consider the internal energy in the plastic hinge from Fig. 7.5 a:

$$\underbrace{W_e}_{\text{external work}} = \underbrace{D}_{\text{dissipation energy} \geq 0} + \underbrace{U}_{\text{elastic energy} = 0} \tag{7.11}$$

where $W_e = W_b$ is the work done by external forces (see Appendix B.1).

The work done by external forces at Sign Convention 1 (see Fig. 7.5 a)

$$\begin{aligned} W_b &= M_{pl}^{*r} \cdot \varphi_l - M_{pl}^{*l} \cdot \varphi_r = |M_{pl}^{*r}| \left[\text{sign} \left(M_{pl}^{*r} \right) \varphi_l - \text{sign} \left(M_{pl}^{*l} \right) \varphi_r \right] \\ &= M_{pl}^* \cdot \Delta^{*I} \varphi = D \geq 0 \end{aligned} \tag{7.12}$$

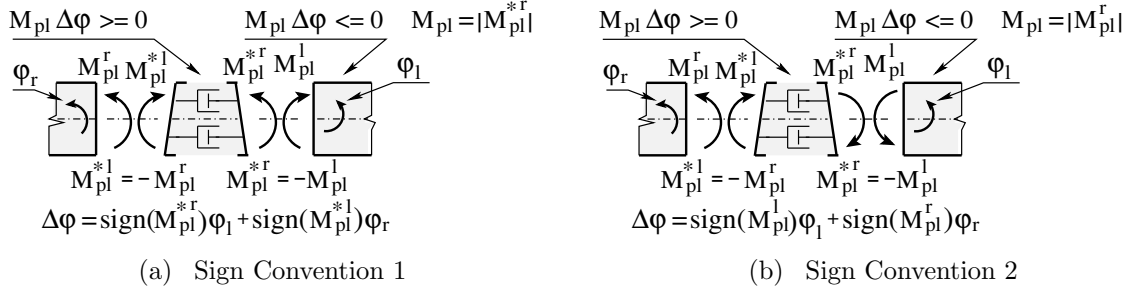


Figure 7.5. Work done in a plastic hinge

The work done by external forces at Sign Convention 2 (see Fig. 7.5 b)

$$\begin{aligned}
 W_b &= -M_{pl}^{*r} \cdot \varphi_l - M_{pl}^{*l} \cdot \varphi_r = |M_{pl}^{*r}| \left[-\text{sign}(M_{pl}^{*r}) \varphi_l - \text{sign}(M_{pl}^{*l}) \varphi_r \right] \\
 &= M_{pl}^* \cdot \Delta^{*II} \varphi = D \geq 0 \quad (7.13)
 \end{aligned}$$

The incremental rotation $\Delta^{II} \varphi$ of the plastic hinge is given by the difference between rotations of the member ends at joint. It is necessary to check the compatibility of the rotations related to the hinge with the sign of the plastic moment of Eq. (7.13).

Considering that $M_{pl}^r = -M_{pl}^{*l}$ and $M_{pl}^l = -M_{pl}^{*r}$, the work of boundary forces at the member ends at joint is

$$W_b = |M_{pl}^r| \cdot \Delta^{II} \varphi = |M_{pl}| \cdot \Delta \varphi = -D \leq 0 \quad (7.14)$$

where

$$\Delta \varphi = \text{sign}(M_{pl}^r) \varphi_r + \text{sign}(M_{pl}^l) \varphi_l \quad (7.15)$$

The *boundary forces work* W_b^r at the member end and the *boundary forces work* W_b^{*l} at the plastic hinge boundary, at a similar angle of rotation, φ_r , have different signs (see Fig. 7.5 b):

$$\begin{aligned}
 W_b^r + W_b^{*l} &= 0 \\
 W_b + W_b^* &= 0 \quad (7.16)
 \end{aligned}$$

The sum of the amounts of work done by internal and external forces is zero (see Appendix B.1 and Eq. (B.8)), or

$$W_e^* = W_b^* = -W_i^* = D \geq 0 \quad (7.17)$$

and

$$W_b = -W_b^* = W_i^* = -D \leq 0 \quad (7.18)$$

The sum of the amounts of work done by boundary forces of members at the plastic hinge

$$W_b = M_{pl}^r \varphi_r + M_{pl}^l \varphi_l = -D \leq 0 \quad (7.19)$$

7.2 Methods of plastic analysis

The criteria to be followed in plastic analysis to identify a correct load factor:

- *equilibrium* – the internal bending moments must be in equilibrium with the external loads;
- *yield* – no point in the structure can have a moment greater than the plastic moment $|M| \leq |M_{pl}|$;
- *mechanism* – at collapse, the structure, or a part of it, can deform as a mechanism;
- *dissipation energy* $D \geq 0$ in Eq. (7.13) must be positive, i.e. the sum of the amounts of work done by boundary forces of members at a plastic hinge $W_b = -D \leq 0$ must be negative (Eq. (7.19)).

Three fundamental theorems are based on the criteria named (see [Cpr11]):

- the lower bound (*safe*) theorem or *static theorem*,
- the upper bound (*unsafe*) theorem or *kinematic theorem*,
- the *uniqueness theorem*.

Theorem 7.2.1 (the lower bound (*safe*) theorem or static theorem)

A load computed on the basis of a bending moment distribution in which the moment nowhere exceeds M_{pl} is either equal to or less than the true collapse load.

A bending moment diagram is found which satisfies the conditions of equilibrium. The load factor λ_i in Eq. (7.1) is either less than or equal to the true load factor at collapse (see Fig. 7.6).

Theorem 7.2.2 (the upper bound (*unsafe*) theorem or kinematic theorem)

A load computed on the basis of an assumed mechanism is either equal to or greater than the true collapse load. When several mechanisms are tried, the true collapse load will be the smallest of them.

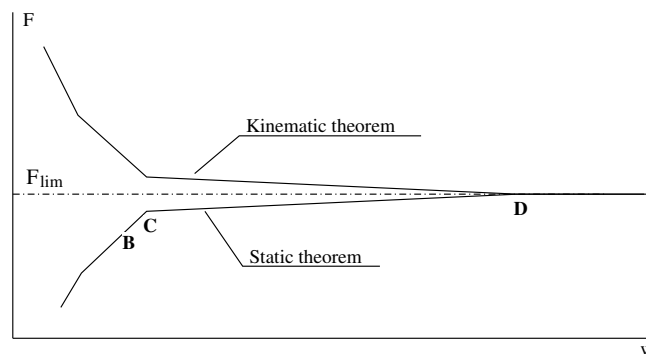


Figure 7.6. Upper and lower bound values

A bending moment diagram is found which satisfies the conditions of equilibrium. The load factor λ_i in Eq. (7.1) is either greater than or equal to the true load factor at collapse (see Fig. 7.6).

Theorem 7.2.3 (the uniqueness theorem)

A load computed on the basis of bending moment distribution which satisfies both the plastic moment and mechanism conditions is a true plastic collapse load.

A bending moment diagram is found which satisfies the conditions of equilibrium. The load factor λ_i of Eq. (7.1) is the true load factor at collapse.

Basically there are three methods of analysis (see [Cpr11] p. 28):

- the kinematic (*mechanism*) method,
- the equilibrium (*statical*) method,
- the load incremental method.

The load incremental method

This method is most readily suited for computer implementation. We will concentrate only on this one, using the EST method (see Example 8.1). Let us assume that

- for an n times statically indeterminate structure, the forces F_i are given,
- the cross-sectional stiffness parameters and the full plastic moment M_{pl} are known.

The method is based on an incremental process producing a sequence of the load factor λ_i of Eq. (7.1). The following steps are made:

1. Increasing the load λ_o times ($\lambda_o F_i$) to the appearance of a full plastic moment $M_{pl}^{(0)}$. For that
 - compose a structural system for an $n-1$ times statically indeterminate structure with a full plastic moment $M_{pl}^{(0)}$ in the plastic hinge,
 - find the load factor λ_o ,
 - check if the boundary work at the plastic hinge satisfies the condition of Eq. (7.19).
2. In an $n-1$ times statically indeterminate structure, increasing the load λ_1 times to the appearance of a second full plastic moment $M_{pl}^{(1)}$. For that
 - compose a structural system for an $n-2$ times statically indeterminate structure with the full plastic moments $M_{pl}^{(0)}$ and $M_{pl}^{(1)}$ in the plastic hinges,
 - find the load factor $\lambda_o + \lambda_1$,

- check if the boundary work at the plastic hinges satisfies the condition of Eq. (7.19) $W_b = M_{pl}^r \varphi_r + M_{pl}^l \varphi_l = -D \leq 0$.
3. Repeating the step-by-step procedure until the structure is statically determinate ($n=0$).
 4. Increasing the load in the statically determinate structure ($n=0$) to the appearance of a full plastic moment $M_{pl}^{(n)}$. A plastic hinge forms and the structure becomes a mechanism. The structure cannot sustain any more load and thus will collapse at F_{lim} . The load computed in which the moment exceeds $M_{pl}^{(n)}$ is equal to the true collapse load. For that
 - compose a structural system for a statically determinate structure with full plastic moments $M_{pl}^{(0)}, M_{pl}^{(1)}, \dots, M_{pl}^{(n)}$ in the plastic hinges,
 - find the load factor $\lambda_0 + \lambda_1 + \dots + \lambda_n$,
 - check if the boundary work at the plastic hinge satisfies the condition of Eq. (7.19) $W_b = M_{pl}^r \varphi_r + M_{pl}^l \varphi_l = -D \leq 0$.

8. The EST method. Limit design

8.1 Illustrative problems

Example 8.1. Problem Statement. The frame depicted in Fig. 8.1 is of height $h = 4$ m and of span length $l = 8$ m. The beam 2-4 is loaded with a vertical concentrated load $F_1 = 1.0\lambda$ kN (here, λ is the load factor, see Eq. (7.1)). A horizontal load $F_2 = 0.5\lambda$ kN acts at joint 2. For the beam, the full plastic moment $M_{pl} = 30.0$ kN·m, and for the columns, $M_{pl} = 15.0$ kN·m.

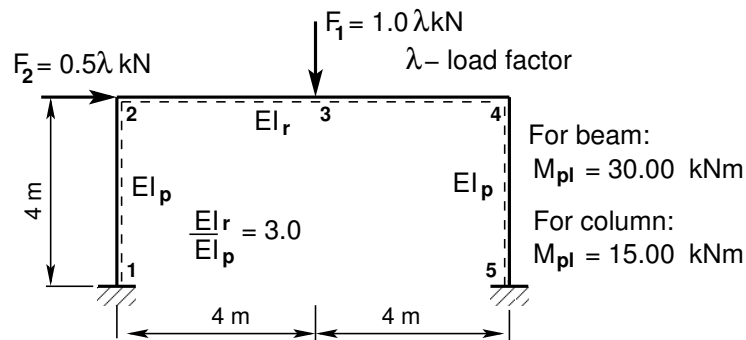


Figure 8.1. The frame ESTsn3

Let us assume that the flexural rigidity of a column $EI_p = 2 \cdot 10^4$ kN·m² and that of the beam $EI_r = 3.0EI_p$; the axial rigidity of a column $EA_p = 4.6 \cdot 10^6$ kN and that of the beam $EA_r = 8.8 \cdot 10^6$ kN; the shear rigidity of a column $GA_{rp} = 0.4EA_p$ and that of the beam $GA_{rr} = 0.4EA_r$.

We wish to compute the collapse load factor λ_C at which the frame will actually fail.

Problem Solving. We use the EST method to solve the problem of an n times statically indeterminate frame. The steps of the load incremental method are listed below.

1. The load of an $n=3$ times statically indeterminate frame is increased λ times ($F_1 = 1.0 \cdot \lambda$, $F_2 = 0.5 \cdot \lambda$, see Fig. 8.1) to the appearance of the full plastic moment M_{pl} . For that
 - compose a structural system for an $n=3$ times statically indeterminate frame (Fig. 8.2 a) with a load factor $\lambda = 1.0$;
 - find the node at which a full plastic moment $M_{plr} = 30.0$ kN·m or $M_{pll} = 15.0$ kN·m will appear. We find that this is node 4, where $M_4 = 1.021$ kN·m (see Fig. 8.2 a);

- find the load factor $\lambda = \lambda_0$ for a moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to appear at node 4. We find that for a full plastic moment to develop at node 4, the load factor is to be $\lambda_0 = 14.685$ as shown in Fig. 8.2b;
 - find the next node at which a full plastic moment $M_{plr} = 30.0 \text{ kN}\cdot\text{m}$ or $M_{pll} = 15.0 \text{ kN}\cdot\text{m}$ will appear. We find that this is node 5 (see Fig. 8.2b). We also find that at this node to achieve $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$, ΔM_5 is needed to add, and that $\Delta M_5 = 15 - 12.2727 = 2.7273 \text{ kN}\cdot\text{m}$ as can be seen in Figs. 8.2b and 8.3a.
2. In an $n=2$ times statically indeterminate frame, load is increased λ ($\lambda = \lambda_0 + \lambda_1$) times to the appearance of the second full plastic moment M_{pl} . For that
- compose a structural system for an $n=2$ times statically indeterminate frame (Fig. 8.3a) with a load factor $\lambda = 1.0$;
 - find the load factor λ_1 for a moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to appear at node 5. We find that for the appearance of a plastic moment at node 5, the load factor is to be $\lambda_1 = \Delta M_5/M_5 = 2.7273/0.8525 = 3.1993$ as shown in Fig. 8.3a;
 - add the boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ (related to λ_0) to the ends of elements 3 and 4 at hinge 4 (see Fig. 8.3b);
 - find the displacements and internal forces of the frame shown in Fig. 8.3b. Here, the load factor is $\lambda = \lambda_0 + \lambda_1 = 17.884$, the corresponding loads are $F_1 = 17.884 \text{ kN}\cdot\text{m}$ and $F_2 = 8.9422 \text{ kN}\cdot\text{m}$;
 - check if the dissipation D at plastic hinge 4 satisfies the condition (7.19). We find that $D_4 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -\left(-7.5000 \cdot 10^{-3} + 8.5190 \cdot 10^{-18}\right) = 0.0075 > 0$;
 - find the next node at which a full plastic moment $M_{plr} = 30.0 \text{ kN}\cdot\text{m}$ or $M_{pll} = 15.0 \text{ kN}\cdot\text{m}$ will form. We find that this is node 3 (see Fig. 8.3b). We also find that at this node, ΔM_3 is needed to add to achieve $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$, and that $\Delta M_3 = 30 - 27.6923 = 2.3077 \text{ kN}\cdot\text{m}$ as shown in Figs. 8.3b and 8.4a.
3. In an $n=1$ times statically indeterminate frame, load is increased λ ($\lambda = \lambda_0 + \lambda_1 + \lambda_2$) times to the appearance of the third full plastic moment M_{pl} . For that
- compose a structural system for an $n=1$ times statically indeterminate frame (Fig. 8.4a) with a load factor $\lambda = 1.0$;
 - find the load factor λ_2 for a moment $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$ to appear at node 3. We find that for the appearance of a plastic moment at node 3, the load factor is to be $\lambda_2 = \Delta M_3/M_3 = 2.3077/2.2727 = 1.0154$ as shown in Fig. 8.4a;
 - add the boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ (related to λ_0) to the ends of elements 3 and 4 at hinge 4 (see Fig. 8.3b);
 - add the boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ (related to λ_1) to the end of element 4 at plastic hinge 5 (see Fig. 8.4b);

- find the displacements and internal forces of the frame shown in Fig. 8.4 b. Here, the load factor is $\lambda = \lambda_0 + \lambda_1 + \lambda_2 = 18.900$, the corresponding loads are $F_1 = 18.900 \text{ kN}\cdot\text{m}$ and $F_2 = 9.450 \text{ kN}\cdot\text{m}$;
- check if the dissipation D at plastic hinges 4 and 5 satisfies the condition (7.19). We find that $D_4 = - (M_{pl}^r \varphi_r + M_{pl}^l \varphi_l) = - (-8.7000 \cdot 10^{-3} - 1.2000 \cdot 10^{-3}) = 0.0099 > 0$ and $D_5 = - (M_{pl}^r \varphi_r) = - (-1.2000 \cdot 10^{-3}) = 0.0012 > 0$;
- find the next node at which a full plastic moment $M_{plr} = 30.0 \text{ kN}\cdot\text{m}$ or $M_{pll} = 15.0 \text{ kN}\cdot\text{m}$ will appear. We find that this is node 1 (see Fig. 8.4 b). We also find that at this node, ΔM_1 is needed to add to achieve $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$, and that $\Delta M_1 = 15 - 8.400 = 6.600 \text{ kN}\cdot\text{m}$ as shown in Figs. 8.4 b and 8.5 a.

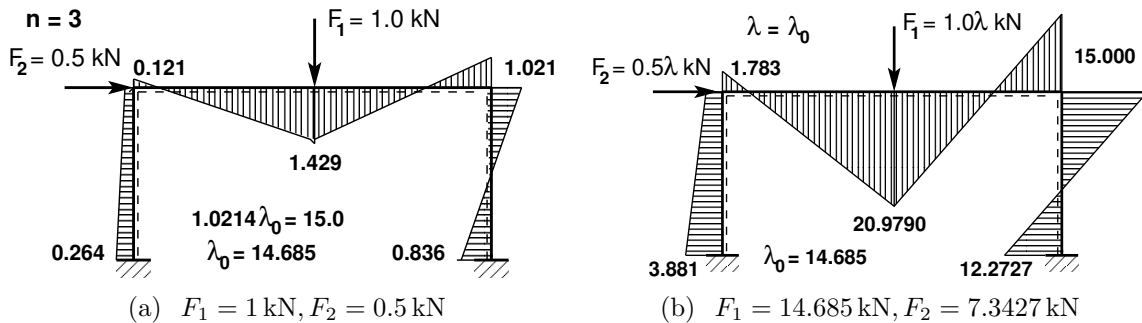


Figure 8.2. Bending moment diagrams of an $n = 3$ times indeterminate frame

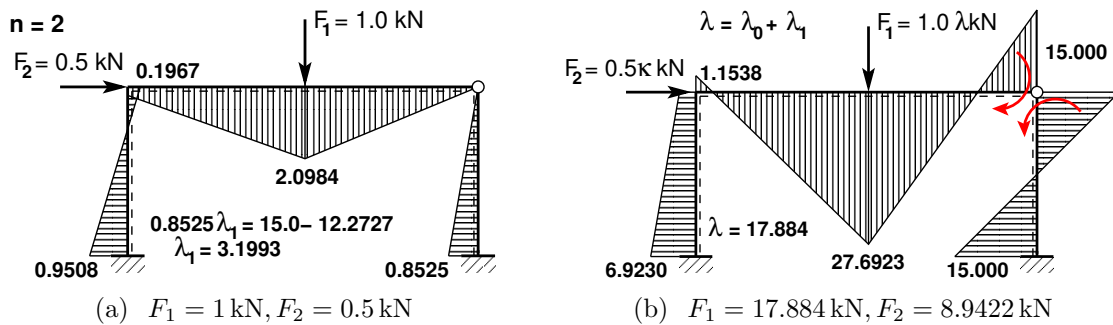


Figure 8.3. Bending moment diagrams of an $n = 2$ times indeterminate frame

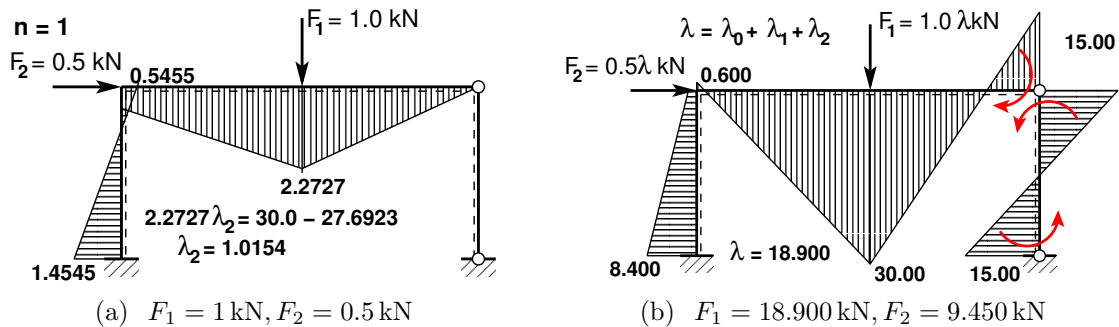


Figure 8.4. Bending moment diagrams of an $n = 1$ times indeterminate frame

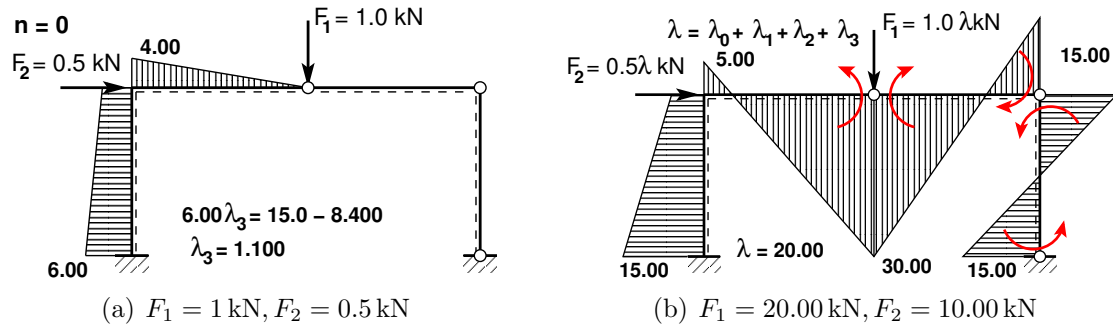


Figure 8.5. Bending moment diagrams of a statically determined frame ($n = 0$)

4. As the last step of the procedure, load is increased in a statically determined frame ($n=0$) to the appearance of the full plastic moment $M_{pl}^{(n)}$. A plastic hinge forms and the frame becomes a mechanism. The load computed in which the moment exceeds $M_{pl}^{(n)}$ is equal to the true collapse load. In the $n=0$ times statically indeterminate (statically determinate) frame, load is increased λ ($\lambda = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3$) times to the appearance of the fourth full plastic moment M_{pl} . For that

- compose a structural system for an $n=0$ times statically indeterminate frame with a load factor $\lambda = 1.0$ (see Fig. 8.5 a);
- find the load factor λ_3 for a moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to develop at node 1. We find that for the appearance of the plastic moment at node 1, the load factor is to be $\lambda_3 = \Delta M_1 / M_1 = 6.600 / 6.000 = 1.1000$ as shown in Fig. 8.5 a;
- add the boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ (related to λ_0) to the ends of elements 3 and 4 at hinge 4 (see Fig. 8.5 b);
- add the boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ (related to λ_1) to the end of element 4 at plastic hinge 5 (see Fig. 8.5 b);
- add the boundary moments that are equal to the full plastic moment $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$ (related to λ_2) to the end of elements 2 and 3 at plastic hinge 3 (see Fig. 8.5 b);
- find the displacements and internal forces of the frame shown in Fig. 8.5 b. Here, the load factor is $\lambda = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 20.000$, the corresponding loads are $F_1 = 20.000 \text{ kN}\cdot\text{m}$ and $F_2 = 10.000 \text{ kN}\cdot\text{m}$;
- check if the dissipation D at plastic hinges 4 and 5 satisfies the condition (7.19). We find that

$$D_4 = - \left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l \right) = - \left(-2.66667 \cdot 10^{-2} - 1.0000 \cdot 10^{-2} \right) = 0.03667 > 0 ,$$

$$D_5 = - \left(M_{pl}^r \varphi_r \right) = - \left(-1.00000 \cdot 10^{-2} \right) = 0.01000 > 0 \text{ and}$$

$$D_3 = - \left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l \right) = - \left(-3.8333 \cdot 10^{-2} - 3.8333 \cdot 10^{-2} \right) = 0.0767 > 0 .$$

The four plastic hinges (nodes 1, 3, 4, and 5) produce a mechanism leading to collapse at the loads $F_1 = 20.000 \text{ kN}\cdot\text{m}$ and $F_2 = 10.000 \text{ kN}\cdot\text{m}$ (related to the load factor $\lambda = 20.000$).

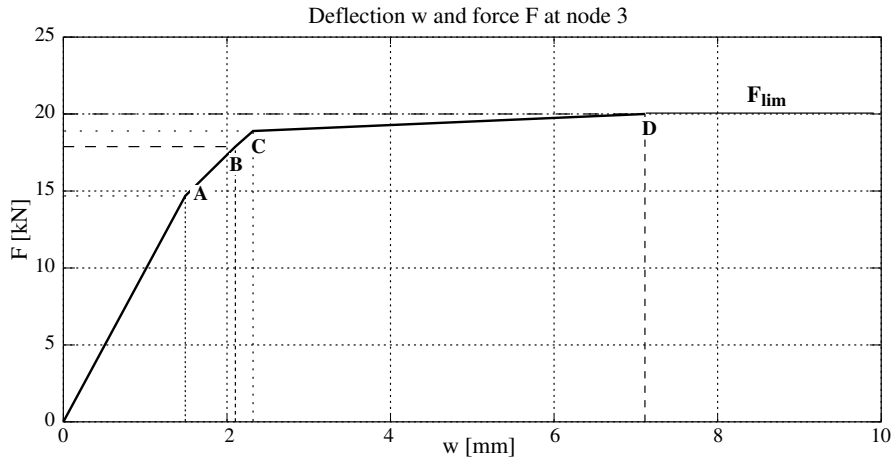


Figure 8.6. Load-deflection relation for the frame ESTsn3

An idealized relation between the load F_2 and deflection at node 3 is displayed in Fig. 8.6. The first yield moment occurs at node 4 shown in Fig. 8.5b and is marked with the letter A in Fig. 8.6. The following yield moments occurring at nodes 5, 3, and 1 shown in Fig. 8.5b are marked with the letters B, C, D in Fig. 8.6.

8.1.1 The $n=3$ times statically indeterminate frame

Example 8.2. Problem Statement. Consider the frame shown in Fig. 8.1 and described in Example 8.1. Our aim is to compute the load factor λ (see Fig. 8.7b) at which the first plastic moment occurs.

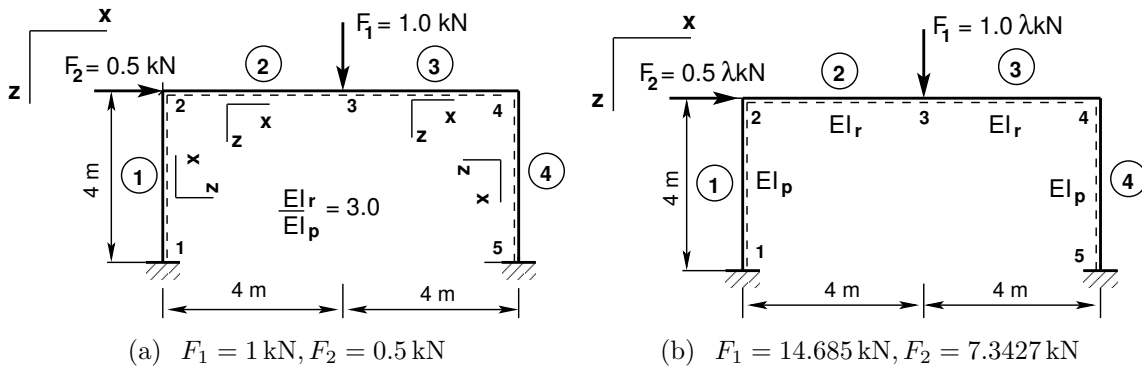


Figure 8.7. Load cases of the $n = 3$ times indeterminate frame ESTsn3

Problem Solving. Now we apply the two load cases shown in Fig. 8.7.

- Load case 1 (Fig. 8.7a). We determine the node at which the first full plastic moment $M_{plr} = 30.0 \text{ kN}\cdot\text{m}$ or $M_{pll} = 15.0 \text{ kN}\cdot\text{m}$ will appear. We find that this is node 4: $M_4 = 1.021 \text{ kN}\cdot\text{m}$ (see Fig. 8.2a). Next we find the load factor $\lambda = \lambda_0$ for a full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to appear at node 4: $\lambda_0 = 14.685$ (see Fig. 8.2a).

- Load case 2 (Fig. 8.7b). We determine the node at which the second full plastic moment $M_{plr} = 30.0 \text{ kN}\cdot\text{m}$ or $M_{pll} = 15.0 \text{ kN}\cdot\text{m}$ will appear. We find that this is node 5 (see Fig. 8.2b). We also find that to achieve $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ at this node, ΔM_5 is needed to add, and that $\Delta M_5 = 15 - 12.2727 = 2.7273 \text{ kN}\cdot\text{m}$ as shown in Figs. 8.2b and 8.3a.

We use the EST method described in Chapter 3, “Statically indeterminate problems”, and carry out the following steps of calculations:

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, side conditions, restrictions on support displacements.

$$\text{spA}\cdot\mathbf{Z} = \mathbf{B} \quad (8.1)$$

2. Assembling and solving the boundary problem equation (8.1) (prepared and solved by the function *LaheFrameSnDFIm.m*):

- (a) writing the basic equations of a frame in transfer matrix form,
- (b) adding the compatibility equations of the displacements at nodes,
- (c) adding the joint equilibrium equations,
- (d) adding the side conditions (moment hinges),
- (e) adding the restrictions on support displacements,
- (f) solving the system of sparse equations,
- (g) producing an output: initial parameter vectors for element displacements and forces; support reactions.

3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTframeSn3LaheWFI.m* are given in excerpts from the program: element and nodal loads – excerpt 8.1; nodal coordinates – excerpt 8.2; element properties, topology and hinges – excerpt 8.3.

Program excerpt 8.1 (*spESTframeSn3LaheWFI.m*)

```

Number_of_frame_nodes=5
Number_of_elements=4
Number_of_support_reactions=6
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts 'Nmitmeks' of the element
Nmitmeks=4
Lp=10.0; % graphic axis
# ---- Load variants ----
load_variant=1;
#load_variant=2;
# --- Element properties ---

```

```

EIp=20000 # kN/m^2
EIr=40000 # kN/m^2
EAp=4.6*10^6
#EAp=4.6*10^15;

EAr=6.8*10^6
#EAr=6.8*10^15;
GAp=0.4*EAp
GAR=0.4*EAr

baasi0=EIp/4 # scaling multiplier for displacements
# baasi0=1.0;
h=4.0;
l=4.0;
l1=1 % 1
l2=1 % 1
Mplr=30.0 # The plastic moment of beam
Mplp=15.0 # The plastic moment of column
# Loads
F2s=0.5;
F3s=1.0;

switch (koormusvariant)
case{1}
disp('--- ')
disp(' Load variant 1 ')
disp('--- ')
#
# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0]; esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0]; esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0; #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s; # 0.5; #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s; # 1.0;

```

```

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0 #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0 #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

case{2}
disp('--- ')
disp(' Load variant 2 ')
disp('--- ')
load myfile30.mat
plastF1F2;
Flambda0=plastF1F2(1,1);
F2s03=plastF1F2(1,2)
F3s03=plastF1F2(1,3)

# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(LoadsqONElement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0]; esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0]; esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0; #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s03; #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s03;

```

```

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0                                #tSiire(:,1,3)= 0.0
#tSiire(2,1,1)= 0.01*baasi0                        #tSiire(:,1,4)= 0.0
#tSiire(:,1,2)= 0.0                                #tSiire(:,1,5)= 0.0

otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 8.2 (spESTframeSn3LaheWFI.m)

```

#=====
#      Nodal coordinates
#=====
krdn=[#  x      z
      0.0    0.0;   % node 1
      0.0   -4.0;   % node 2
      4.0   -4.0;   % node 3
      8.0   -4.0;   % node 4
      8.0    0.0]; % node 5
#=====

#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#=====
tsolm=[1      1 1 1; % node 1
      5      1 1 1]; % node 5
#=====

```

Program excerpt 8.3 (spESTframeSn3LaheWFI.m)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#          n2 - end of the element
#          n1 - beginning of the element
#          N, Q, M - hinges at the end of the element
#          N, Q, M - hinges at the beginning of the element
#          Mpl - plastic moment at the end of the element
#          Mpl - plastic moment at the beginning
#                                     of the element
EIp EAp GAp  2    1    0 0 0    0 0 0    0.0 0.0 ;   % element 1
EIr EAr GAR  3    2    0 0 0    0 0 0    0.0 0.0 ;   % element 2
EIr EAr GAR  4    3    0 0 0    0 0 0    0.0 0.0 ;   % element 3
EIp EAp GAp  5    4    0 0 0    0 0 0    0.0 0.0 ]; % element 4
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (8.1), carried out by the function `LaheFrameSnDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. This function gives the unscaled initial parameter vectors of the elements and support reactions according to the load cases.*

3. Output data.

Load case 1. *The load factor $\lambda = \lambda_0$ (`Flambda0`), forces F_1 (`F3s03`) and F_2 (`F2s03`) are given in excerpt 8.1 from the computing diary.*

Computing diary excerpt 8.1 (`spESTframeSn3LaheWFI.m`)

```
Flambda0=15/AbsmaxminM
Flambda0 = 14.685
-----
F2s = 0.50000
F2s03=Flambda0*F2s
F2s03 = 7.3427
---
F3s = 1
F3s03=Flambda0*F3s
F3s03 = 14.685
=====
plastF1F2=[Flambda0 F2s03 F3s03]
plastF1F2 =

    14.6853    7.3427    14.6853

save myfile30.mat plastF1F2
```

The bending moments of the first loading case are given in excerpt 8.2 from the computing diary, and the bending moments diagram is shown in Fig. 8.2 a.

Computing diary excerpt 8.2 (`spESTframeSn3LaheWFI.m`)

```
Moments at nodes (Sign Convention 2)
=====
```

Node	Moment	DaF No	Fi/baasi0	DaF No	Work_b	Element
1	0.26429	12	0.00000e+00	9	0.00000e+00	1
2	-0.12143	6	-3.85714e-05	3	4.68367e-06	1
2	0.12143	24	-3.85714e-05	21	-4.68367e-06	2
3	1.42857	18	5.00000e-06	15	7.14286e-06	2
3	-1.42857	36	5.00000e-06	33	-7.14286e-06	3
4	-1.02143	30	1.85714e-05	27	-1.89694e-05	3
4	1.02143	48	1.85714e-05	45	1.89694e-05	4
5	0.83571	42	0.00000e+00	39	0.00000e+00	4

```
-----
```

Note. Work_b is done by boundary forces of an element

Load case 2. *The bending moments of the second loading case are given in excerpt 8.4 of the computing diary. The bending moments diagram is shown in Fig. 8.2 b. We find that the next node at which a full plastic moment $M_{pl} = 15.0$ kN·m will appear is node 5 ($DaF = 42$, see computing diary excerpt 8.3, Figs. 8.8 and 8.2 b). An this node, $\Delta M_5 = 15 - 12.2727 = 2.7273$ kN·m (ΔM_0) is to be added as shown in excerpt 8.3 and Fig. 8.3 a.*

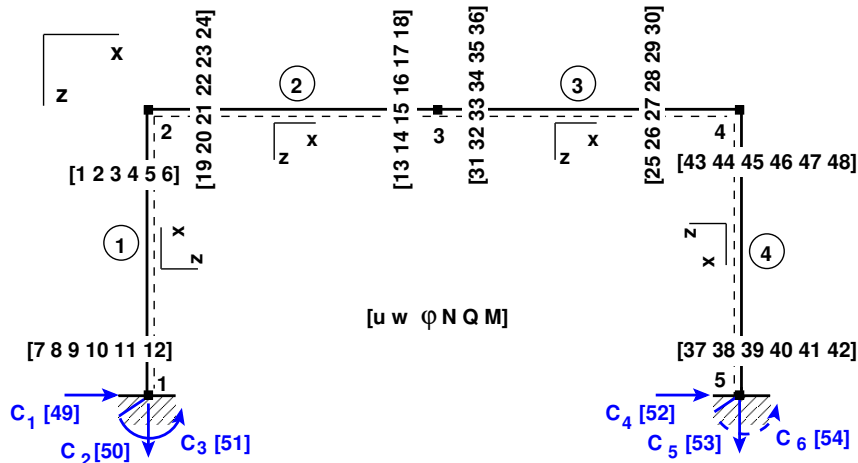


Figure 8.8. Numeration of displacements and forces of the frame ESTsn3

Computing diary excerpt 8.3 (spESTframeSn3LaheWFI.m)

```
The maximum absolute values of reduced moments
AbsmaxminM = 12.273
muutujaDaF1 = 42
deltaM0=15.0-AbsmaxminM
deltaM0 = 2.7273
Flambda0 = 14.685
deltaM0 = 2.7273
muutujaDaF1 = 42
save myfile31.mat Flambda0 deltaM0 muutujaDaF1
```

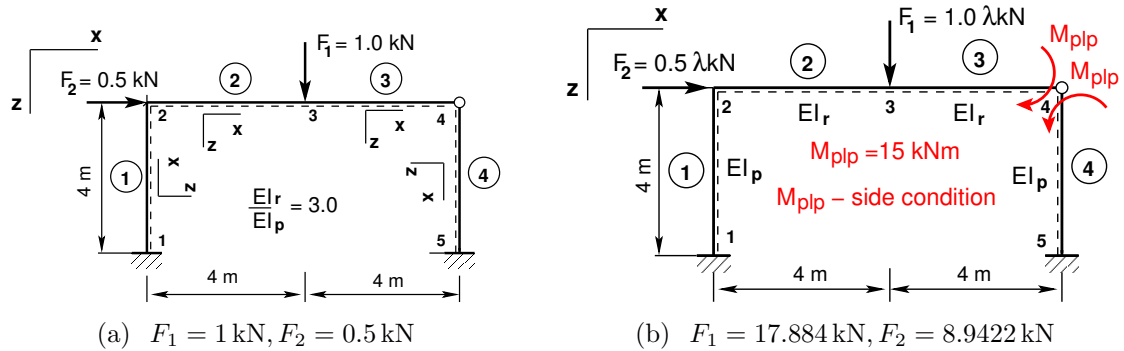
Computing diary excerpt 8.4 (spESTframeSn3LaheWFI.m)

Moments at nodes (Sign Convention 2)

Node	Moment	DaF No	Fi/baasi0	DaF No	Work_b	Element
1	3.88112	12	0.00000e+00	9	0.00000e+00	1
2	-1.78322	6	-5.66434e-04	3	1.01007e-03	1
2	1.78322	24	-5.66434e-04	21	-1.01007e-03	2
3	20.97902	18	7.34266e-05	15	1.54042e-03	2
3	-20.97902	36	7.34266e-05	33	-1.54042e-03	3
4	-15.00000	30	2.72727e-04	27	-4.09091e-03	3
4	15.00000	48	2.72727e-04	45	4.09091e-03	4
5	12.27273	42	0.00000e+00	39	0.00000e+00	4

8.1.2 The $n=2$ times statically indeterminate frame

Example 8.3. Problem Statement. Consider the frame depicted in Fig. 8.1 and described in Example 8.1. We wish to compute the load factor λ (see Fig. 8.9b) at which the second plastic moment occurs.

Figure 8.9. Load cases of the $n = 2$ times indeterminate frame ESTsn2

Problem Solving. The two load cases shown in Fig. 8.9 are applied:

- Load case 1 (Fig. 8.9 a). The load factor λ_1 for a moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to appear at node 5 must be found. We find that for a plastic moment to appear at node 5, the load factor is to be $\lambda_1 = \Delta M_5 / M_5 = 2.7273 / 0.8525 = 3.1993$ as shown in Fig. 8.3 a (ΔM_5 was found on page 187).
- Load case 2 (Fig. 8.9 b). The boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ are added to the ends of elements 3 and 4 at plastic hinge 4 (see excerpt 8.6). The corresponding load factor, $\lambda = \lambda_0 + \lambda_1$, is shown in Fig. 8.3 b. We find now that at node 3, a third full plastic moment $M_{pl,r} = 30.0 \text{ kN}\cdot\text{m}$ appears (see Fig. 8.3 b). We also find that ΔM_3 is needed to add at this node to achieve $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$, and that $\Delta M_3 = 30 - 27.692 = 2.3077 \text{ kN}\cdot\text{m}$ as shown in Figs. 8.3 b and 8.4 a.

We use the EST method described in Chapter 3, “Statically indeterminate problems”, and carry out the following steps of calculations:

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, side conditions, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (8.2)$$

2. Assembling and solving the boundary problem equation (8.2) (prepared and solved by the function `LaheFrameSnDFIm.m`):
 - (a) writing the basic equations of a frame in transfer matrix form,
 - (b) adding the compatibility equations of the displacements at nodes,
 - (c) adding the joint equilibrium equations,
 - (d) adding the side conditions (moment hinges),
 - (e) adding the restrictions on support displacements,
 - (f) solving the system of sparse equations,
 - (g) producing an output: initial parameter vectors for element displacements and forces; support reactions.
3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTframeSn2LaheWFI.m* are given in excerpts from the program: element and nodal loads – excerpt 8.4; nodal coordinates – excerpt 8.5; element properties, topology and hinges – excerpt 8.6.

Program excerpt 8.4 (spESTframeSn2LaheWFI.m)

```

Number_of_frame_nodes=5
Number_of_elements=4
Number_of_support_reactions=6
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts ''Nmitmeks'' of the element
Nmitmeks=4
Lp=10.0; % graphic axis
# ---- Load variants -----
load_variant=1;
#load_variant=2;
# --- Element properties ---

EIp=20000 # kN/m^2          EAr=6.8*10^6
EIr=40000 # kN/m^2        #EAr=6.8*10^15;
EAp=4.6*10^6              GAp=0.4*EAp
#EAp=4.6*10^15;          GAr=0.4*EAr

baasi0=EIp/4 # scaling multiplier for displacements
# baasi0=1.0;
h=4.0;
l=4.0;
l1=1 % 1
l2=1 % 1
Mplr=30.0 # The plastic moment of beam
Mplp=15.0 # The plastic moment of column
# Loads
F2s=0.5;
F3s=1.0;
switch (koormusvariant)
case{1}
disp('--- ')
disp(' Load variant 1 ')
disp('--- ')
#
disp(' myfile31.mat Flambda0 deltaM0 muutujaDaF1 ')
disp('load myfile31.mat ')
load myfile31.mat
#
Flambda0
deltaM0
muutujaDaF1
#
# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;

```

```

esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions  kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];              esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0];              esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0;                          #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s; # 0.5;                      #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s; # 1.0;

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0                          #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0                  #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

case{2}
disp('--- ')
disp(' Load variant 2 ')
disp('--- ')
load myfile31.mat
Flambda0
disp(' load myfile20.mat ')
load myfile20.mat
disp(' plastF1F2; ')
#plastF1F2=[Flambda Flambda0 Flambda1 F2s02 F3s02]
plastF1F2;
Flambda=plastF1F2(1,1)
Flambda0=plastF1F2(1,2)
Flambda1=plastF1F2(1,3)
F2s02=plastF1F2(1,4)
F3s02=plastF1F2(1,5)

```

```

# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];      esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0];      esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0;      #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s02;      #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s02;

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0      #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0      #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 8.5 (`spESTframeSn2LahWFI.m`)

```

#=====
# Nodal coordinates
#=====
krdn=[# x z
      0.0 0.0; % node 1

```

```

0.0   -4.0;   % node 2
4.0   -4.0;   % node 3
8.0   -4.0;   % node 4
8.0   0.0];   % node 5
#=====
#
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No   u   w   fi
#=====
tsolm=[1       1 1 1; % node 1
       5       1 1 1]; % node 5
#=====

```

Program excerpt 8.6 (`spESTframeSn2LaheWFI.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#         n2 - end of the element
#         n1 - beginning of the element
#         N, Q, M - hinges at the end of the element
#         N, Q, M - hinges at the beginning of the element
#         Mpl - plastic moment at the end of the element
#         Mpl - plastic moment at the beginning
#         of the element
EIp EAp GAp  2   1   0 0 0   0 0 0   0.0 0.0 ;   % element 1
EIr EAr GAR  3   2   0 0 0   0 0 0   0.0 0.0 ;   % element 2
EIr EAr GAR  4   3   0 0 1   0 0 0  -Mplp 0.0 ;   % element 3
EIp EAp GAp  5   4   0 0 0   0 0 1   0.0 Mplp ]; % element 4
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (8.2), carried out by the function `LaheFrameSnDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. This function gives the unscaled initial parameter vectors of the elements and support reactions according to the load cases.*

3. Output data.

Load case 1. The load factor $\lambda = \lambda_0 + \lambda_1$ (`Flambda`), and forces F_1 (`F3s02`) and F_2 (`F2s02`) are presented in excerpt 8.5 from the computing diary.

Computing diary excerpt 8.5 (`spESTframeSn2LaheWFI.m`)

```

deltaM0=15.0-AbsmaxminM
deltaM0 = 2.7273
-----
absX = 0.85246
Flambda1=deltaM0/absX
Flambda1 = 3.1993
Flambda=Flambda0+Flambda1
Flambda = 17.885
-----
F2s02=Flambda*F2s
F2s02 = 8.9423
F3s02=Flambda*F3s

```

```

F3s02 = 17.885
=====
plastF1F2=[Flambda Flambda0 Flambda1 F2s02 F3s02]
plastF1F2 =

    17.8846    14.6853     3.1993     8.9423    17.8846

save myfile20.mat plastF1F2

```

The bending moments of the first loading case are presented in excerpt 8.2 from the computing diary, and the bending moments diagram is shown in Fig. 8.3 a.

Computing diary excerpt 8.6 (`spESTframeSn2LaheWFI.m`)

Moments at nodes (Sign Convention 2)

```

=====
Node      Moment DaF No  Fi/baasi0  DaF No  Work_b      Element
-----
1         0.95082   12  0.00000e+00   9  0.00000e+00   1
2         0.19672    6 -7.54098e-05   3 -1.48347e-05   1
2        -0.19672   24 -7.54098e-05  21  1.48347e-05   2
3         2.09836   18  1.09290e-06  15  2.29329e-06   2
3        -2.09836   36  1.09290e-06  33 -2.29329e-06   3
4         0.00000   30  7.10383e-05  27  0.00000e+00   3
4         0.00000   48 -8.52459e-05  45 -0.00000e+00   4
5         0.85246   42  0.00000e+00  39  0.00000e+00   4
-----

```

Load case 2. The bending moments of the second loading case are given in excerpt 8.8 from the computing diary, and the bending moments diagram is shown in Fig. 8.3 b. We find that the next node at which a full plastic moment $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$ will appear is node 3 (DaF = 36, see computing diary excerpt 8.7, Figs. 8.8 and 8.3 b). At this node, $\Delta M_3 = 30 - 27.6923 = 2.3077 \text{ kN}\cdot\text{m}$ (deltaM0) is to be added as shown in excerpt 8.7 and Fig. 8.4 a.

Computing diary excerpt 8.7 (`spESTframeSn2LaheWFI.m`)

The maximum absolute values of reduced moments

```

AbsmaxminM = 27.6923
Flambda = 17.885
Flambda0 = 14.685
Flambda1 = 3.1993
deltaM0 = 2.3077
muutujaDaF2 = 36
save myfile21.mat Flambda Flambda0 Flambda1 deltaM0 muutujaDaF2

```

Computing diary excerpt 8.8 (`spESTframeSn2LaheWFI.m`)

Moments at nodes (Sign Convention 2)

```

=====
Node      Moment DaF No  Fi/baasi0  DaF No  Work_b      Element
-----
1         6.92308   12  0.00000e+00   9  0.00000e+00   1
2        -1.15385    6 -8.07692e-04   3  9.31953e-04   1
2         1.15385   24 -8.07692e-04  21 -9.31953e-04   2

```

3	27.69231	18	7.69231e-05	15	2.13018e-03	2
3	-27.69231	36	7.69231e-05	33	-2.13018e-03	3
4	-15.00000	30	5.00000e-04	27	-7.50000e-03	3
4	15.00000	48	5.67933e-19	45	8.51899e-18	4
5	15.00000	42	0.00000e+00	39	0.00000e+00	4

At node 4, the dissipation

$$D = -(-7.50000e-03 + 8.51899e-18) = 0.00750 > 0$$

The dissipation D at plastic hinge 4 satisfies the condition (7.19):

$$D_4 = -(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l) = -(-7.5000 \cdot 10^{-3} + 8.5190 \cdot 10^{-18}) = 0.0075 \text{ kN}\cdot\text{m} > 0.$$

8.1.3 The $n=1$ times statically indeterminate frame

Example 8.4. Problem Statement. Let us consider the frame shown in Fig. 8.1 and described in Example 8.1. We wish to compute the load factor λ (see Fig. 8.10 b) at which the third plastic moment occurs.

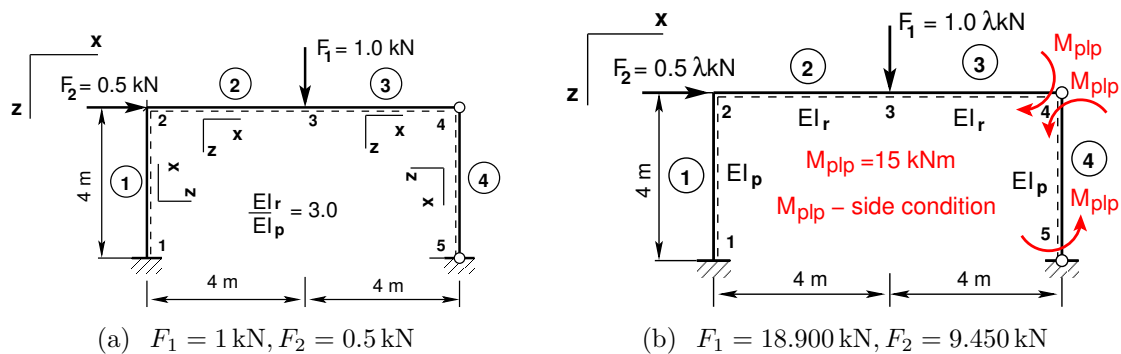


Figure 8.10. Load cases of the $n = 1$ times indeterminate frame ESTsn1

Problem Solving. The two load cases shown in Fig. 8.10 are applied:

- Load case 1 (Fig. 8.10 a). We determine the load factor λ_2 for a moment $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$ to appear at node 3. We find that for a plastic moment to form at node 5, the load factor is to be $\lambda_2 = \Delta M_3 / M_3 = 2.3077 / 2.2727 = 1.0154$ as shown in Fig. 8.4 a (ΔM_3 (deltaM0) was found on page 193).
- Load case 2 (Fig. 8.10 b). The boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ are added to the ends of elements 3 and 4 at plastic hinges 4 and 5 (see excerpt 8.9). The corresponding load factor, $\lambda = \lambda_0 + \lambda_1 + \lambda_2$, is shown in Fig. 8.4 b. We find that at node 1, a fourth full plastic moment, $M_{plr} = 15.0 \text{ kN}\cdot\text{m}$, will appear (see Fig. 8.4 b). We also find that at this node ΔM_1 is needed to add to achieve $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ and that $\Delta M_1 = 15 - 8.4000 = 6.6000 \text{ kN}\cdot\text{m}$ as shown in Figs. 8.4 b and 8.5 a.

We use the EST method described in Chapter 3, “Statically indeterminate problems”, and carry out the following steps of calculations:

1. *Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, side conditions, restrictions on support displacements.*

$$\mathbf{spA} \cdot \mathbf{Z} = \mathbf{B} \quad (8.3)$$

2. *Assembling and solving the boundary problem equation (8.3) (prepared and solved by the function `LaheFrameSnDFIm.m`):*
 - (a) *writing the basic equations of a frame in transfer matrix form,*
 - (b) *adding the compatibility equations of the displacements at nodes,*
 - (c) *adding the joint equilibrium equations,*
 - (d) *adding the side conditions (moment hinges),*
 - (e) *adding the restrictions on support displacements,*
 - (f) *solving the system of sparse equations,*
 - (g) *producing an output: initial parameter vectors for element displacements and forces; support reactions.*
3. *Output: element displacements and forces determined by the transfer matrix.*

1. **Input data** for the GNU Octave program `spESTframeSn1LaheWFI.m` are given in excerpts from the program: element and nodal loads – excerpt 8.7; nodal coordinates – excerpt 8.8; element properties, topology and hinges – excerpt 8.9.

Program excerpt 8.7 (`spESTframeSn1LaheWFI.m`)

```

Number_of_frame_nodes=5
Number_of_elements=4
Number_of_support_reactions=5
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts 'Nmitmeks' of the element
Nmitmeks=4
Lp=10.0; % graphic axis
# ---- Load variants ----
load_variant=1;
#load_variant=2;
# --- Element properties ---

EIp=20000 # kN/m^2           EAr=6.8*10^6
EIr=40000 # kN/m^2         #EAr=6.8*10^15;
EAp=4.6*10^6               GAp=0.4*EAp
#EAp=4.6*10^15;           GAR=0.4*EAr

baasi0=EIp/4 # scaling multiplier for displacements
# baasi0=1.0;
h=4.0;
l=4.0;
l1=1 % 1
l2=1 % 1

```

```

Mplr=30.0 # The plastic moment of beam
Mlp=15.0 # The plastic moment of column
# Loads
F2s=0.5;
F3s=1.0;

switch (koormusvariant)
case{1}
disp('--- ')
disp(' Load variant 1 ')
disp('--- ')
#
disp('deltaM0=15.0-AbsmaxminM ')
# disp('load myfile21.mat Flambda Flambda0 Flambda1 deltaM0 muutuajaDaF2')
disp('load myfile21.mat ')
load myfile21.mat
#
Flambda
Flambda0
Flambda1
deltaM0
muutuajaDaF2
#
# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];      esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0];      esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0;      #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s; # 0.5;      #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s; # 1.0;

```

```

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0                                #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0                        #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

case{2}
disp('--- ')
disp(' Load variant 2 ')
disp('--- ')
disp(' load myfile10.mat ')
load myfile10.mat
disp(' plastF1F2; ')%
#plastF1F2=[Flambda Flambda0 Flambda1 Flambda2 F2s01 F3s01]
plastF1F2;
Flambda=plastF1F2(1,1)
Flambda0=plastF1F2(1,2)
Flambda1=plastF1F2(1,3)
Flambda2=plastF1F2(1,4)
F2s01=plastF1F2(1,5)
F3s01=plastF1F2(1,6)

# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];              esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0];              esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

```

```

#sSolmF(:,1,1)= 0.0;          #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s01;        #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s01;

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0          #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0 #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 8.8 (`spESTframeSn1LaheWFI.m`)

```

#=====
#      Nodal coordinates
#=====
krdn=[#  x      z
      0.0    0.0;   % node 1
      0.0   -4.0;   % node 2
      4.0   -4.0;   % node 3
      8.0   -4.0;   % node 4
      8.0    0.0]; % node 5
#=====

# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#=====
tsolm=[1      1 1 1; % node 1
       5      1 1 0]; % node 5
#=====

```

Program excerpt 8.9 (`spESTframeSn1LaheWFI.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#         n2 - end of the element
#         n1 - beginning of the element
#         N, Q, M - hinges at the end of the element
#         N, Q, M - hinges at the beginning of the element
#         Mpl - plastic moment at the end of the element
#         Mpl - plastic moment at the beginning
#         of the element

```

```

EIp EAp GAp 2 1 0 0 0 0 0 0 0.0 0.0 ; % element 1
EIr EAr GAr 3 2 0 0 0 0 0 0 0.0 0.0 ; % element 2
EIr EAr GAr 4 3 0 0 1 0 0 0 -Mplp 0.0 ; % element 3
EIp EAp GAp 5 4 0 0 1 0 0 1 Mplp Mplp ]; % element 4
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (8.3), carried out by the function `LaheFrameSnDFIm(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. This function gives the unscaled initial parameter vectors of the elements and support reactions according to the load cases.*

3. Output data.

Load case 1. The load factor $\lambda = \lambda_0 + \lambda_1 + \lambda_2$ (`Flambda`), forces F_1 (`F3s01`) and F_2 (`F2s01`) are presented in excerpt 8.9 from the computing diary.

Computing diary excerpt 8.9 (`spESTframeSn1LaheWFI.m`)

```

deltaM0 = 2.3077
-----
absX = 2.2727
Flambda2=deltaM0/absX
Flambda2 = 1.0154
Flambda=Flambda0+Flambda1+Flambda2
Flambda = 18.900
-----
F2s01=Flambda*F2s
F2s01 = 9.4500
F3s01=Flambda*F3s
F3s01 = 18.900
=====
plastF1F2=[Flambda Flambda0 Flambda1 Flambda2 F2s01 F3s01]
plastF1F2 =

    18.9000    14.6853     3.1993     1.0154     9.4500    18.9000

save myfile10.mat plastF1F2

```

The bending moments of the first loading case are given in excerpt 8.2 from the computing diary, and the bending moments diagram is shown in Fig. 8.3 a.

Computing diary excerpt 8.10 (`spESTframeSn1LaheWFI.m`)

Moments at nodes (Sign Convention 2)

```

=====
Node      Moment DaF No  Fi/baasi0  DaF No  Work_b  Element
-----
1         1.45455   12  0.00000e+00  9  0.00000e+00  1
2         0.54545    6 -9.09091e-05  3 -4.95868e-05  1
2        -0.54545   24 -9.09091e-05  21  4.95868e-05  2
3         2.27273   18  3.03030e-06  15  6.88705e-06  2
3        -2.27273   36  3.03030e-06  33 -6.88705e-06  3
4         0.00000   30  7.87879e-05  27  0.00000e+00  3
4         0.00000   48 -7.87879e-05  45 -0.00000e+00  4
5         0.00000   42 -7.87879e-05  39 -0.00000e+00  4
=====

```

Load case 2. The bending moments of the second loading case are given in excerpt 8.12 from the computing diary, and the bending moments diagram is shown in Fig. 8.4 b. We find that the next node for a full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to form at is node 1 ($DaF = 12$, see excerpt 8.11 from the computing diary, Figs. 8.8 and 8.4 b). At this node, $\Delta M_1 = 15 - 8.4000 = 6.6000 \text{ kN}\cdot\text{m}$ (deltaM0) is needed to add as shown in excerpt 8.11 and Fig. 8.4 a.

Computing diary excerpt 8.11 (spESTframeSn1LaheWFI.m)

The maximum absolute values of reduced moments

AbsmaxminM = 8.4000

muutujaDaF3 = 12

AbsmaxminM = 8.4000

Flambda = 18.900

Flambda0 = 14.685

Flambda1 = 3.1993

Flambda2 = 1.0154

deltaM0 = 6.6000

muutujaDaF3 = 12

save myfile11.mat Flambda Flambda0 Flambda1 Flambda2 deltaM0 muutujaDaF3

Computing diary excerpt 8.12 (spESTframeSn1LaheWFI.m)

Moments at nodes (Sign Convention 2)

```
=====
```

Node	Moment	DaF No	Fi/baasi0	DaF No	Work_b	Element
1	8.40000	12	0.00000e+00	9	0.00000e+00	1
2	-0.60000	6	-9.00000e-04	3	5.40000e-04	1
2	0.60000	24	-9.00000e-04	21	-5.40000e-04	2
3	30.00000	18	8.00000e-05	15	2.40000e-03	2
3	-30.00000	36	8.00000e-05	33	-2.40000e-03	3
4	-15.00000	30	5.80000e-04	27	-8.70000e-03	3
4	15.00000	48	-8.00000e-05	45	-1.20000e-03	4
5	15.00000	42	-8.00000e-05	39	-1.20000e-03	4

```
=====
```

At node 4, the dissipation

$$D = -(-8.70000e-03 - 1.20000e-03) = 0.00990 > 0$$

At node 5, the dissipation

$$D = -(-1.20000e-03) = 0.00120 > 0$$

The dissipation D at plastic hinges 4 and 5 satisfies the condition (7.19):

$$D_4 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -(-8.70000e - 03 - 1.20000e - 03) = 0.00990 \text{ kN}\cdot\text{m} > 0,$$

$$D_5 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -(-1.20000e - 03) = 0.00120 \text{ kN}\cdot\text{m} > 0.$$

8.1.4 The $n=0$ times statically indeterminate frame

Example 8.5. Problem Statement. Let us consider the frame shown in Fig. 8.1 and described in Example 8.1. We wish to compute the load factor λ (see Fig. 8.11 b) at which the fourth plastic moment occurs.

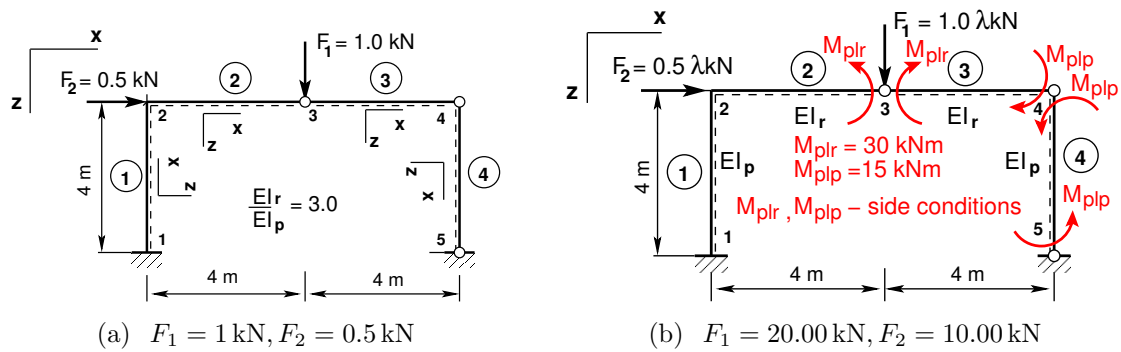


Figure 8.11. Load cases of the $n = 0$ times indeterminate frame ESTsn0

Problem Solving. The two load cases depicted in Fig. 8.11 are applied:

- Load case 1 (see Fig. 8.11 a). We determine the load factor λ_3 needed for a moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ to appear at node 1. We find that a plastic moment will form at node 1 if the load factor $\lambda_3 = \Delta M_1 / M_1 = 6.6000 / 6.0000 = 1.1000$ as shown in Fig. 8.5 a ($\Delta M_1 = 15 - 8.4000 = 6.6000 \text{ kN}\cdot\text{m}$ (deltaM0) was found on page 200).
- Load case 2 (see Fig. 8.11 b). The boundary moments that are equal to the full plastic moment $M_{pl} = 15.0 \text{ kN}\cdot\text{m}$ are added to the ends of elements 3 and 4 at plastic hinges 4 and 5. Similarly attached to the ends of elements 2 and 3 at plastic hinge 3, are the boundary moments that are equal to the full plastic moment $M_{pl} = 30.0 \text{ kN}\cdot\text{m}$ (see excerpt 8.12). The corresponding load factor, $\lambda = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3$, is shown in Fig. 8.5 b. We now find that at node 1, a fourth full plastic moment $M_{plr} = 15.0 \text{ kN}\cdot\text{m}$ forms (see Fig. 8.5 b).

Making use of the EST method described in Chapter 3, “Statically indeterminate problems”, we carry out the following steps of calculations:

1. Data input: the number of frame nodes, elements, support reactions; element properties, element loads in local coordinates, node forces in global coordinates, nodal coordinates, topology and hinges, side conditions, restrictions on support displacements.

$$\text{spA} \cdot \mathbf{Z} = \mathbf{B} \tag{8.4}$$

2. Assembling and solving the boundary problem equation (8.4) (prepared and solved by the function `LaheFrameSnDFIm.m`):
 - (a) writing the basic equations of a frame in transfer matrix form,
 - (b) adding the compatibility equations of the displacements at nodes,
 - (c) adding the joint equilibrium equations,
 - (d) adding the side conditions (moment hinges),
 - (e) adding the restrictions on support displacements,
 - (f) solving the system of sparse equations,
 - (g) producing an output: initial parameter vectors for element displacements and forces; support reactions.
3. Output: element displacements and forces determined by the transfer matrix.

1. **Input data** for the GNU Octave program *spESTframeSn0LaheWFI.m* are given in excerpts from the program: element and nodal loads – excerpt 8.10; nodal coordinates – excerpt 8.11; element properties, topology and hinges – excerpt 8.12.

Program excerpt 8.10 (spESTframeSn0LaheWFI.m)

```

Number_of_frame_nodes=5
Number_of_elements=4
Number_of_support_reactions=5
spNNK=12*Number_of_elements+Number_of_support_reactions;
Number_of_unknowns=spNNK
Displacements and forces are calculated on parts ''Nmitmeks'' of the element
Nmitmeks=4
Lp=10.0; % graphic axis
# ---- Load variants -----
load_variant=1;
#load_variant=2;
# --- Element properties ---

EIp=20000 # kN/m^2           EAr=6.8*10^6
EIr=40000 # kN/m^2         #EAr=6.8*10^15;
EAp=4.6*10^6               GAr=0.4*EAr
#EAp=4.6*10^15;           GAR=0.4*EAr

baasi0=EIp/4 # scaling multiplier for displacements
# baasi0=1.0;
h=4.0;
l=4.0;
l1=1 % 1
l2=1 % 1
Mplr=30.0 # The plastic moment of beam
Mplp=15.0 # The plastic moment of column
# Loads
F2s=0.5;
F3s=1.0;

switch (koormusvariant)

case{1}
disp('--- ')
disp(' Load variant 1 ')
disp('--- ')
#
disp('deltaM0=15.0-AbsmaxminM ')
disp(' myfile11.mat Flambda Flambda0 Flambda1 Flambda2 deltaM0 muutuajaDaF3 ')
disp('load myfile11.mat ')
load myfile11.mat
#
Flambda
Flambda0
Flambda1
Flambda2
deltaM0

```



```

muutuJaDaF3
#
# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0Nelement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0];      esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0];      esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0];      esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0;      #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s; # 0.5;      #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s; # 1.0;

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0      #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0      #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

case{2}
disp('--- ')
disp(' Load variant 2 ')
disp('--- ')
disp(' load myfile00.mat ')
load myfile00.mat
disp(' plastF1F2; ')
#disp('plastF1F2=[Flambda Flambda0 Flambda1 Flambda2 Flambda3 F2s00 F3s00] ')

```

```

plastF1F2;
Flambda=plastF1F2(1,1)
Flambda0=plastF1F2(1,2)
Flambda1=plastF1F2(1,3)
Flambda2=plastF1F2(1,4)
Flambda3=plastF1F2(1,5)
F2s00=plastF1F2(1,6)
F3s00=plastF1F2(1,7)
#

# Element load in local coordinates
# qz qx qA qL
# Uniformly distributed load in local coordinate z and x directions
Loadsq on element=4;
esQkoormus=zeros(Loadsq0NElement,4,ElementideArv);

esQkoormus(1,1:4,1)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,3)=[0.0 0.0 0.0 4.0];
esQkoormus(1,1:4,2)=[0.0 0.0 0.0 4.0]; esQkoormus(1,1:4,4)=[0.0 0.0 0.0 4.0];

#
# Point load in local coordinate z and x directions kN
# Fz, Fx, aF (coordinate of the point of force application)
LoadsF on Element=5;
esFjoud=zeros(LoadsF_on_Element,2,ElementideArv);

esFjoud(1,1:3,1)=[0.0 0.0 4.0]; esFjoud(1,1:3,3)=[0.0 0.0 4.0];
esFjoud(1,1:3,2)=[0.0 0.0 4.0]; esFjoud(1,1:3,4)=[0.0 0.0 4.0];
#esFjoud(2,1:3,2)=[0.0 0.0 4.0];

#
# Node forces in global coordinates
# sSolmF(forces,1,nodes); forces=[Fx; Fz; My]
sSolmF = zeros(3,1,SolmedeArv);

#sSolmF(:,1,1)= 0.0; #sSolmF(:,1,4)= 0.0
sSolmF(1,1,2)= F2s00; #sSolmF(:,1,5)= 0.0
sSolmF(2,1,3)= F3s00;

#
#s1F(1,1,1)=0.0; # force Fz
#s1F(2,1,1)=0.0; # force Fz
#s1F(3,1,1)=0.0; # force My

# Support shift - tSiire#
# Support shift is multiplied by scaling multiplier
tSiire = zeros(3,1,SolmedeArv);

#tSiire(:,1,1)= 0.0 #tSiire(:,1,4)= 0.0
#tSiire(2,1,1)= 0.01*baasi0 #tSiire(:,1,5)= 0.0
#tSiire(:,1,2)= 0.0
#tSiire(:,1,3)= 0.0

```

```

otherwise
disp(' No load variant cases ')
endswitch

```

Program excerpt 8.11 (`spESTframeSn0LaheWFI.m`)

```

=====
#      Nodal coordinates
=====
krdn=[#  x      z
      0.0    0.0;   % node 1
      0.0   -4.0;   % node 2
      4.0   -4.0;   % node 3
      8.0   -4.0;   % node 4
      8.0    0.0]; % node 5
=====
#
#=====
# Restrictions on support displacements (on - 1, off - 0)
# Support No  u  w  fi
#=====
tsolm=[1      1 1 1; % node 1
      5      1 1 0]; % node 5
=====

```

Program excerpt 8.12 (`spESTframeSn0LaheWFI.m`)

```

# ----- Element properties, topology and hinges -----
elasts=[# Element properties
#          n2 - end of the element
#          n1 - beginning of the element
#          N, Q, M - hinges at the end of the element
#          N, Q, M - hinges at the beginning of the element
#          Mpl - plastic moment at the end of the element
#          Mpl - plastic moment at the beginning
#          of the element
EIp EAp GAp  2    1    0 0 0    0 0 0    0.0  0.0 ;    % element 1
EIr EAr GAR  3    2    0 0 0    0 0 1    Mplr 0.0 ;    % element 2
EIr EAr GAR  4    3    0 0 1    0 0 1   -Mplr -Mplr ;  % element 3
EIp EAp GAp  5    4    0 0 1    0 0 1    Mplp Mplp ]; % element 4
# 1 - hinge 'true' (axial, shear, moment hinges)
#

```

2. *Assembling and solving the boundary problem equations (8.4), carried out by the function `LaheFrameSnDFIm(basi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`. This function gives the unscaled initial parameter vectors of the elements and support reactions according to the load cases.*

3. Output data.

Load case 1. The load factor $\lambda = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3$ (*Flambda*), forces F_1 (*F3s00*) and F_2 (*F2s00*) are shown in excerpt 8.13 from the computing diary.

Computing diary excerpt 8.13 (spESTframeSn0LaheWFI.m)

```

muutujaDaF3 = 12
-----
deltaM0 = 6.6000
-----
absX = 6
Flambda3=deltaM0/absX
Flambda3 = 1.1000
Flambda=Flambda0+Flambda1+Flambda2+Flambda3
Flambda = 20
-----
F2s01=Flambda*F2s
F2s00 = 10
F3s01=Flambda*F3s
F3s00 = 20
=====
plastF1F2=[Flambda Flambda0 Flambda1 Flambda2 Flambda3 F2s00 F3s00]
plastF1F2 =

    20.0000    14.6853     3.1993     1.0154     1.1000    10.0000    20.0000

save myfile00.mat plastF1F2

```

The bending moments in the first loading case are given in excerpt 8.2 from the computing diary, and the bending moments diagram is shown in Fig. 8.3 a.

Computing diary excerpt 8.14 (spESTframeSn0LaheWFI.m)

Moments at nodes (Sign Convention 2)

```

=====
Node      Moment DaF No  Fi/baasi0  DaF No  Work_b  Element
-----
1         6.00000    12  0.00000e+00  9  0.00000e+00  1
2        -4.00000     6 -1.00000e-03  3  4.00000e-03  1
2         4.00000    24 -1.00000e-03  21 -4.00000e-03  2
3         0.00000    18 -1.13333e-03  15 -0.00000e+00  2
3         0.00000    36  1.08889e-03  33  0.00000e+00  3
4         0.00000    30  1.08889e-03  27  0.00000e+00  3
4         0.00000    48 -5.33333e-04  45 -0.00000e+00  4
5         0.00000    42 -5.33333e-04  39 -0.00000e+00  4
-----

```

Load case 2. The bending moments in the second loading case are given in excerpt 8.15 from the computing diary, and now the bending moments diagram is shown in Fig. 8.5 b.

Computing diary excerpt 8.15 (spESTframeSn0LaheWFI.m)

Moments at nodes (Sign Convention 2)

```

=====
Node      Moment DaF No  Fi/baasi0  DaF No  Work_b  Element
-----
1        15.00000    12  0.00000e+00  9  0.00000e+00  1
2        -5.00000     6 -2.00000e-03  3  1.00000e-02  1
2         5.00000    24 -2.00000e-03  21 -1.00000e-02  2

```

3	30.00000	18	-1.16667e-03	15	-3.50000e-02	2
3	-30.00000	36	1.27778e-03	33	-3.83333e-02	3
4	-15.00000	30	1.77778e-03	27	-2.66667e-02	3
4	15.00000	48	-6.66667e-04	45	-1.00000e-02	4
5	15.00000	42	-6.66667e-04	39	-1.00000e-02	4

 At node 4, the dissipation

$$D = -(-3.83333e-02 - 3.83333e-02) = 0.07667 > 0$$

At node 4, the dissipation

$$D = -(-2.66667e-02 - 1.00000e-02) = 0.03667 > 0$$

At node 5, the dissipation

$$D = -(-1.00000e-02) = 0.01000 > 0$$

The dissipation D at plastic hinges 4 and 5 satisfies the condition (7.19):

$$D_3 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -(-3.83333e-02 - 3.83333e-02) = 0.07667 \text{ kN}\cdot\text{m} > 0,$$

$$D_4 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -(-2.66667e-02 - 1.00000e-02) = 0.03667 \text{ kN}\cdot\text{m} > 0,$$

$$D_5 = -\left(M_{pl}^r \varphi_r + M_{pl}^l \varphi_l\right) = -(-1.00000e-02) = 0.01000 \text{ kN}\cdot\text{m} > 0.$$

We have found the collapse load factor $\lambda_C = \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 20.000$ at which the frame will actually fail (see Fig. 8.5 b).

Part IV
Appendices

A. Matrices

A matrix type that stores only the values of non-zero elements and their row and column indexes is generally called sparse^{1 2}. For the storage and creation of sparse matrices we use *GNU Octave*^{3 4}.

A.1 Sparse matrices and GNU Octave

A.1.1 Introduction to sparse matrices

For calculating support reactions and interaction forces on statically determinate hinged beams, also known as Gerber^{5 6} beams (see Fig. A.2), we have a system of equilibrium equations where the coefficient matrix is sparse. The sparsity pattern of this matrix spA is shown in Fig. A.1.

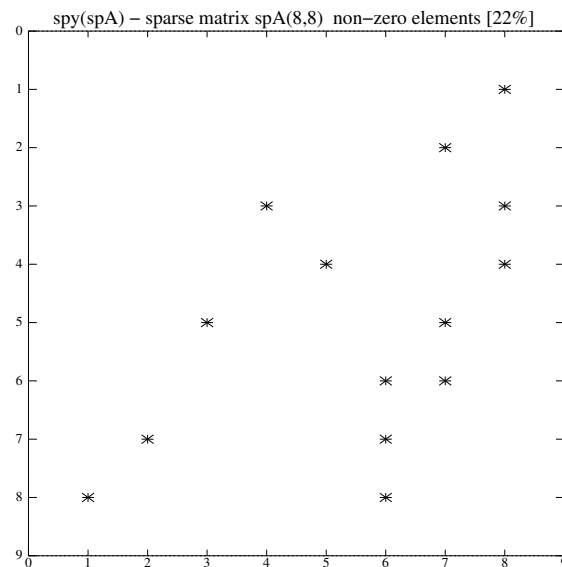


Figure A.1. Sparsity pattern of matrix spA

¹<http://www.gnu.org/software/octave/doc/interpreter/Sparse-Matrices.html>. Web. 08 August 2013.

²http://en.wikipedia.org/wiki/Sparse_matrix. Web. 08 August 2013.

³http://www.obihiro.ac.jp/~suzukim/masuda/octave/html3/octave_112.html#SEC216. Web. 08 August 2013.

⁴http://www.network-theory.co.uk/docs/octave3/octave_205.html. Web. 08 August 2013.

⁵ Heinrich Gerber (1832–1912), a German civil engineer and inventor of multispan hinged beams.

⁶http://de.wikipedia.org/wiki/Heinrich_Gottfried_Gerber. Web. 08 August 2013.

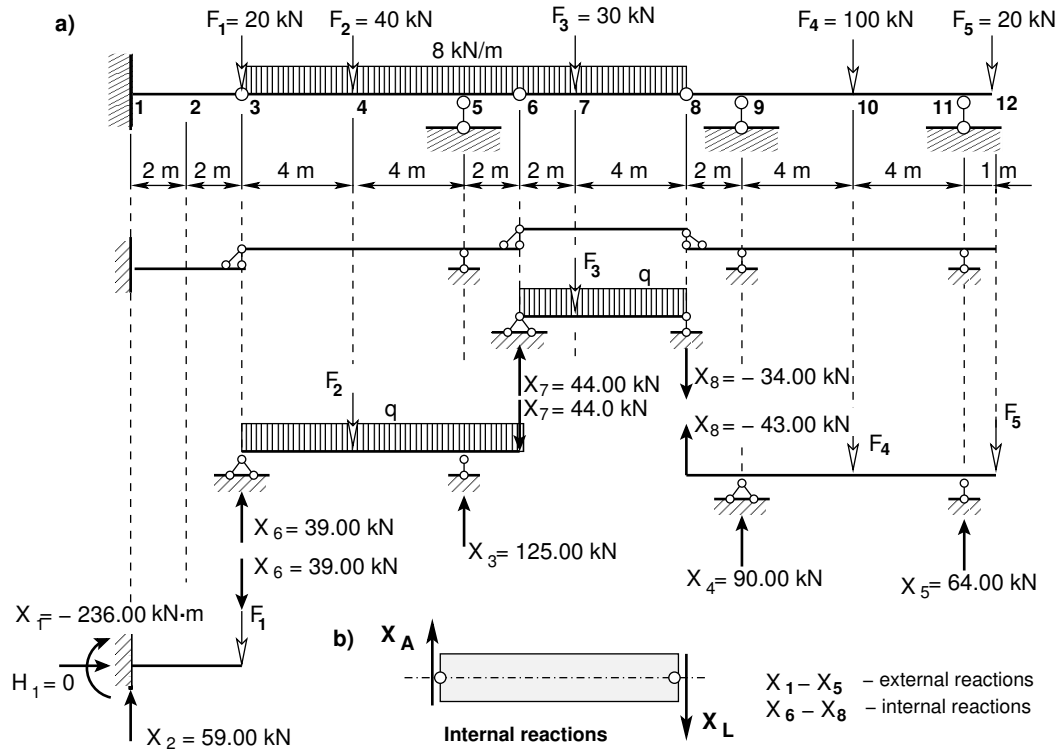


Figure A.2. Assembly sequence of a Gerber beam

To prevent a large number of calculations, this system is decomposed in such a way that an unknown force could be calculated directly with each equilibrium equation.

Consider the equilibrium equations for the beams in Fig. A.2:

beam 6–8

$$\Sigma M_6 = 0; \quad X_8 \cdot 6 + F_3 \cdot 2 + q \cdot 6 \cdot 3 = 0 \quad (\text{A.1})$$

$$\Sigma M_8 = 0; \quad -X_7 \cdot 6 + F_3 \cdot 4 + q \cdot 6 \cdot 3 = 0 \quad (\text{A.2})$$

beam 8–12

$$\Sigma M_{11} = 0; \quad -X_8 \cdot 10 - X_4 \cdot 8 + F_4 \cdot 4 - F_5 \cdot 1 = 0 \quad (\text{A.3})$$

$$\Sigma M_9 = 0; \quad -X_8 \cdot 2 + X_5 \cdot 8 - F_4 \cdot 4 - F_5 \cdot 9 = 0 \quad (\text{A.4})$$

beam 3–6

$$\Sigma M_3 = 0; \quad -X_7 \cdot 10 + X_3 \cdot 8 - F_2 \cdot 4 - q \cdot 10 \cdot 5 = 0 \quad (\text{A.5})$$

$$\Sigma M_5 = 0; \quad -X_7 \cdot 2 - X_6 \cdot 8 + F_2 \cdot 4 + q \cdot 10 \cdot (5 - 2) = 0 \quad (\text{A.6})$$

beam 1–3

$$\Sigma Z = 0; \quad X_6 - X_2 + F_1 = 0 \quad (\text{A.7})$$

$$\Sigma M_1 = 0; \quad X_1 + X_6 \cdot 4 + F_1 \cdot 4 = 0 \quad (\text{A.8})$$

Now rewrite the systems of equations (A.1)–(A.8) in matrix form:

$$\begin{array}{l}
 1. \Sigma M_6 = 0; \\
 2. \Sigma M_8 = 0; \\
 3. \Sigma M_{11} = 0; \\
 4. \Sigma M_9 = 0; \\
 5. \Sigma M_3 = 0; \\
 6. \Sigma M_5 = 0; \\
 7. \Sigma Z = 0; \\
 8. \Sigma M_1 = 0
 \end{array}
 \begin{bmatrix}
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 6 \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & -6 & \cdot \\
 \cdot & \cdot & \cdot & -8 & \cdot & \cdot & \cdot & -10 \\
 \cdot & \cdot & \cdot & \cdot & 8 & \cdot & \cdot & -2 \\
 \cdot & \cdot & 8 & \cdot & \cdot & \cdot & -10 & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & -8 & -2 & \cdot \\
 \cdot & -1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
 1 & \cdot & \cdot & \cdot & \cdot & 4 & \cdot & \cdot
 \end{bmatrix}
 \begin{bmatrix}
 X_1 \\
 X_2 \\
 X_3 \\
 X_4 \\
 X_5 \\
 X_6 \\
 X_7 \\
 X_8
 \end{bmatrix}
 =
 \begin{bmatrix}
 F_3 \cdot 2 + q \cdot 6 \cdot 3 \\
 F_3 \cdot 4 + q \cdot 6 \cdot 3 \\
 F_4 \cdot 4 - F_5 \cdot 1 \\
 -F_4 \cdot 4 - F_5 \cdot 9 \\
 -F_2 \cdot 4 - q \cdot 10 \cdot 5 \\
 F_2 \cdot 4 + q \cdot 10 \cdot (5 - 2) \\
 F_1 \\
 F_1 \cdot 4
 \end{bmatrix}
 \quad (\text{A.9})$$

We have obtained the sparse system (8×8) of equilibrium equations

$$\mathbf{A} \cdot \mathbf{X} = -\mathbf{B} \quad (\text{A.10})$$

where

$$\mathbf{A} = \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 \\
 0 & 0 & 0 & 0 & 0 & 0 & -6 & 0 \\
 0 & 0 & 0 & -8 & 0 & 0 & 0 & -10 \\
 0 & 0 & 0 & 0 & 8 & 0 & 0 & -2 \\
 0 & 0 & 8 & 0 & 0 & 0 & -10 & 0 \\
 0 & 0 & 0 & 0 & 0 & -8 & -2 & 0 \\
 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 4 & 0 & 0
 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}
 -204 \\
 -264 \\
 -380 \\
 580 \\
 560 \\
 -400 \\
 -20 \\
 -80
 \end{bmatrix} \quad (\text{A.11})$$

The Gerber beam support reactions are calculated by hand in the reverse order to that of the assembly sequence (see calculating order in Eqs. (A.1)–(A.9), and Fig. A.2). The sparsity pattern of the matrix \mathbf{A} of Eq. (A.11) is shown in Fig. A.1.

The non-zero elements of the matrix \mathbf{A} can be represented as the row (\mathbf{iv}), column (\mathbf{jv}) and data (\mathbf{sv}) vectors in Eq. (A.12).

$$\begin{array}{l}
 \mathbf{iv} = [1 \quad 2 \quad 3 \quad 3 \ 4 \quad 4 \ 5 \quad 5 \quad 6 \quad 6 \quad 7 \ 7 \ 8 \ 8] \\
 \mathbf{jv} = [8 \quad 7 \quad 4 \quad 8 \ 5 \quad 8 \ 3 \quad 7 \quad 6 \quad 7 \quad 2 \ 6 \ 1 \ 6] \\
 \mathbf{sv} = [6 \quad -6 \quad -8 \quad -10 \ 8 \quad -2 \ 8 \quad -10 \quad -8 \quad -2 \quad -1 \ 1 \ 1 \ 4]
 \end{array} \quad (\text{A.12})$$

A.1.2 Creating sparse matrices

GNU Octave uses the compressed column format *storage technique*.⁷ There are several modes to create a sparse matrix. Sparse matrices can be constructed from matrices or vectors.

The function `sparse(iv, jv, sv)` has constructed a sparse matrix `spA` (shown in computing diary A.1) from three vectors representing the row (`iv`), column (`jv`) and data (`sv`) (see Eq. (A.12)). In this diary, the conversion of the sparse matrix `spA` to a full matrix `A` (`full(spA)`) is shown.

Computing diary A.1

```
octave:1> iv = [1 2 3 3 4 4 5 5 6 6 7 7 8 8]
iv =

    1    2    3    3    4    4    5    5    6    6    7    7    8    8

octave:2> jv = [8 7 4 8 5 8 3 7 6 7 2 6 1 6]
jv =

    8    7    4    8    5    8    3    7    6    7    2    6    1    6

octave:3> sv = [6 -6 -8 -10 8 -2 8 -10 -8 -2 -1 1 1 4]
sv =

    6   -6   -8  -10    8   -2    8  -10   -8   -2   -1    1    1    4

octave:4> spA=sparse(iv, jv, sv)
spA =
Compressed Column Sparse (rows = 8, cols = 8, nnz = 14 [22%])

(8, 1) -> 1          (8, 6) -> 4
(7, 2) -> -1         (2, 7) -> -6
(5, 3) -> 8          (5, 7) -> -10
(3, 4) -> -8         (6, 7) -> -2
(4, 5) -> 8          (1, 8) -> 6
(6, 6) -> -8         (3, 8) -> -10
(7, 6) -> 1          (4, 8) -> -2

octave:5> A=full(spA)
A =
    0    0    0    0    0    0    0    6
    0    0    0    0    0    0   -6    0
    0    0    0   -8    0    0    0  -10
    0    0    0    0    8    0    0   -2
    0    0    8    0    0    0  -10    0
    0    0    0    0    0   -8   -2    0
    0   -1    0    0    0    1    0    0
    1    0    0    0    0    4    0    0

octave:6>
```

⁷http://www.obihiro.ac.jp/~suzukim/masuda/octave/html3/octave_113.html#SEC219.
Web. 08 August 2013.

The non-zero elements of the matrix \mathbf{A} of Eq. (A.11) can be represented as the matrix $\mathbf{A1}$ with columns \mathbf{iv} , \mathbf{jv} and \mathbf{sv} , see Eq. (A.12):

$$\mathbf{A1} = \begin{bmatrix} 1 & 8 & 6 \\ 2 & 7 & -6 \\ 3 & 4 & -8 \\ 3 & 8 & -10 \\ 4 & 5 & 8 \\ 4 & 8 & -2 \\ 5 & 3 & 8 \\ 5 & 7 & -10 \\ 6 & 6 & -8 \\ 6 & 7 & -2 \\ 7 & 2 & -1 \\ 7 & 6 & 1 \\ 8 & 1 & 1 \\ 8 & 6 & 4 \end{bmatrix} \quad (\text{A.13})$$

The function `spconvert(A1)` (given in computing diary A.2) constructs a sparse matrix `spA` from the matrix $\mathbf{A1}$.

Computing diary A.2

```

octave:1> A1=[1 8 6;
2 7 -6;
3 4 -8;
3 8 -10;
4 5 8;
4 8 -2;
5 3 8;
5 7 -10;
6 6 -8;
6 7 -2;
7 2 -1;
7 6 1;
8 1 1;
8 6 4]
A1 =
1 8 6
2 7 -6
3 4 -8
3 8 -10
4 5 8
4 8 -2
5 3 8
5 7 -10
6 6 -8
6 7 -2
7 2 -1
7 6 1
8 1 1
8 6 4

octave:2> spA=spconvert(A1)
spA =
Compressed Column Sparse (rows = 8, cols = 8, nnz = 14 [22%])

(8, 1) -> 1
(7, 2) -> -1
(5, 3) -> 8
(3, 4) -> -8
(4, 5) -> 8
(6, 6) -> -8
(7, 6) -> 1
(8, 6) -> 4
(2, 7) -> -6
(5, 7) -> -10
(6, 7) -> -2
(1, 8) -> 6
(3, 8) -> -10
(4, 8) -> -2

```

The function `sparse(A)` converts the full matrix **A** to a sparse matrix (see computing diary [A.3](#)).

Computing diary A.3

```
octave:6> A
A =
    0    0    0    0    0    0    0    6
    0    0    0    0    0    0   -6    0
    0    0    0   -8    0    0    0  -10
    0    0    0    0    8    0    0   -2
    0    0    8    0    0    0  -10    0
    0    0    0    0    0   -8   -2    0
    0   -1    0    0    0    1    0    0
    1    0    0    0    0    4    0    0

octave:7> spA=sparse(A)
spA =
Compressed Column Sparse (rows = 8, cols = 8, nnz = 14 [22%])
    (8, 1) -> 1          (8, 6) -> 4
    (7, 2) -> -1         (2, 7) -> -6
    (5, 3) -> 8          (5, 7) -> -10
    (3, 4) -> -8         (6, 7) -> -2
    (4, 5) -> 8          (1, 8) -> 6
    (6, 6) -> -8         (3, 8) -> -10
    (7, 6) -> 1          (4, 8) -> -2

octave:8>
```

A.1.3 Sparse matrix functions in the EST method

There are several functions that manipulate sparse matrices: `full`, `sparse`, `spconvert`, `spfind`, `sprank`, `spy`, `speye`, etc. ⁸

We shall introduce the GNU Octave function `spA=spInsertBtoA(spA,M,N,spB)` (p. 241) written for the EST method. This function inserts a sparse matrix `spB` into the sparse matrix `spA`, starting at row index `M` and column index `N`. The overlapping elements of the matrices `spA` and `spB` are added together.

The insertion of the matrix **B** (Eq. (A.14)) into the sparse matrix **spC** is described in computing diary [A.4](#). There, the elements of the matrix **C** of value 2 ($C(5,5)$ and $C(6,6)$) have been obtained as the sum of overlapped elements of matrices `spB` and `spB1`.

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.14})$$

⁸http://www.obihiro.ac.jp/~suzukim/masuda/octave/html3/octave_113.html#SEC222.
Web. 08 August 2013.

Computing diary A.4

```
octave:1> iv = [1 2 3 4 5 6]
```

```
iv =
```

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

```
octave:2> jv = [1 2 4 3 5 6]
```

```
jv =
```

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

```
octave:3> sv = [1 1 1 1 1 1]
```

```
sv =sparse matrix spB
```

```
  1  1  1  1  1  1
```

```
octave:4> spB=sparse(iv, jv, sv)
```

```
spB =
```

```
Compressed Column Sparse (rows = 6, cols = 6, nnz = 6 [17%])
```

```
(1, 1) -> 1
```

```
(2, 2) -> 1
```

```
(4, 3) -> 1
```

```
(3, 4) -> 1
```

```
(5, 5) -> 1
```

```
(6, 6) -> 1
```

```
octave:5> spB1=spB;
```

```
octave:6> spC=sparse(10,10)
```

```
spC =
```

```
Compressed Column Sparse (rows = 10, cols = 10, nnz = 0 [0%])
```

```
octave:7> spC=spInsertBtoA(spC,1,1,spB);
```

```
octave:8> spC=spInsertBtoA(spC,5,5,spB1)
```

```
spC =
```

```
Compressed Column Sparse (rows = 10, cols = 10, nnz = 10 [10%])
```

```
(1, 1) -> 1
```

```
(2, 2) -> 1
```

```
(4, 3) -> 1
```

```
(3, 4) -> 1
```

```
(5, 5) -> 2
```

```
(6, 6) -> 2
```

```
(8, 7) -> 1
```

```
(7, 8) -> 1
```

```
(9, 9) -> 1
```

```
(10, 10) -> 1
```

```
octave:9> C=full(spC)
C =
```

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1

```
octave:10>
```

Next we introduce the GNU Octave function `spA=spSisestaArv(spA,M,N,V)` (p. 242) written for the EST method. This function inserts a numeric value **V** into the sparse matrix **spA** at row index **M** and column index **N**.

A.2 Transformation matrices

Consider the two right-handed coordinate systems of Fig. A.3, defined by orthogonal unit vectors **i, j, k**, and **i*, j*, k***. Let *xyz* be global coordinates and *x*y*z** a local coordinate system.

The vector \vec{F} in Fig. A.3 can be written as the sum of two vectors along the coordinate axes **i, k** with magnitude F_x, F_z and along the coordinate axes **i*, k*** with magnitude F_x^*, F_z^* .

$$\vec{F} = F_x \cdot \vec{i} + F_z \cdot \vec{k} = F_x^* \vec{i}^* + F_z^* \vec{k}^*, \quad \left\{ \begin{array}{c} \vec{i}^* \\ \vec{k}^* \end{array} \right\}, \quad \left\{ \begin{array}{c} \vec{i} \\ \vec{k} \end{array} \right\} \quad (\text{A.15})$$

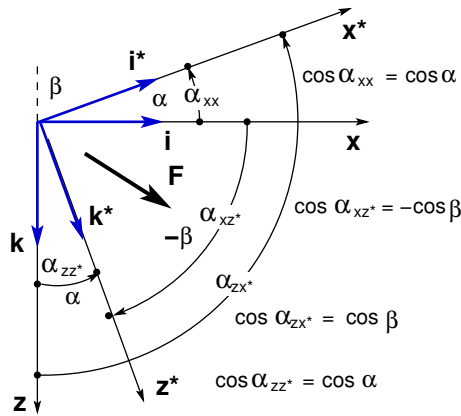


Figure A.3. Coordinate transformation

To find the components of the vector $\vec{\mathbf{F}}$ of Eq. (A.15), we multiply this equation by $\vec{\mathbf{i}}^*$ and $\vec{\mathbf{k}}^*$. The scalar products are:

$$\begin{aligned}\vec{\mathbf{F}} \cdot \vec{\mathbf{i}}^* &= F_x^* = F_x \cdot \vec{\mathbf{i}} \cdot \vec{\mathbf{i}}^* + F_z \cdot \vec{\mathbf{k}} \cdot \vec{\mathbf{i}}^* \\ \vec{\mathbf{F}} \cdot \vec{\mathbf{k}}^* &= F_z^* = F_x \cdot \vec{\mathbf{i}} \cdot \vec{\mathbf{k}}^* + F_z \cdot \vec{\mathbf{k}} \cdot \vec{\mathbf{k}}^*\end{aligned}\tag{A.16}$$

where the scalar product of the two orthogonal vectors is zero.

To find the inverse transformation, we multiply Eq. (A.15) by $\vec{\mathbf{i}}$ and $\vec{\mathbf{k}}$. The scalar products are:

$$\begin{aligned}\vec{\mathbf{F}} \cdot \vec{\mathbf{i}} &= F_x = F_x^* \cdot \vec{\mathbf{i}}^* \cdot \vec{\mathbf{i}} + F_z^* \cdot \vec{\mathbf{k}}^* \cdot \vec{\mathbf{i}} \\ \vec{\mathbf{F}} \cdot \vec{\mathbf{k}} &= F_z = F_x^* \cdot \vec{\mathbf{i}}^* \cdot \vec{\mathbf{k}} + F_z^* \cdot \vec{\mathbf{k}}^* \cdot \vec{\mathbf{k}}\end{aligned}\tag{A.17}$$

The scalar product of the two unit vectors is related to the cosine of the angle between these vectors (Fig. A.3).

$$\begin{aligned}\vec{\mathbf{i}} \cdot \vec{\mathbf{i}}^* &= \vec{\mathbf{i}}^* \cdot \vec{\mathbf{i}} = \cos \alpha_{xx^*}, & \vec{\mathbf{i}} \cdot \vec{\mathbf{k}}^* &= \cos \alpha_{xz^*} \\ \vec{\mathbf{k}} \cdot \vec{\mathbf{k}}^* &= \vec{\mathbf{k}}^* \cdot \vec{\mathbf{k}} = \cos \alpha_{zz^*}, & \vec{\mathbf{i}}^* \cdot \vec{\mathbf{k}} &= \cos \alpha_{zx^*}\end{aligned}\tag{A.18}$$

In Fig. A.3, we show the direction cosines of the vector $\vec{\mathbf{F}}$:

$$\begin{aligned}\cos \alpha_{xx^*} &= \cos \alpha, & \cos \alpha_{zx^*} &= \cos \beta \\ \cos \alpha_{zz^*} &= \cos \alpha, & \cos \alpha_{xz^*} &= -\cos \beta\end{aligned}\tag{A.19}$$

Be careful using cosine and sine angles associated with coordinate system transformation: $\cos \alpha_{xx^*} = \cos \alpha$ and $\cos \alpha_{zx^*} = \cos \beta$ ($\cos \beta = \cos (90^\circ + \alpha) = -\sin \alpha$).

The length l and the direction cosines of an element can be calculated using coordinates x_A, z_A of the node at the beginning and coordinates x_L, z_L of the node at the end of the element (Fig. A.4):

$$l = \sqrt{(z_L - z_A)^2 + (x_L - x_A)^2}\tag{A.20}$$

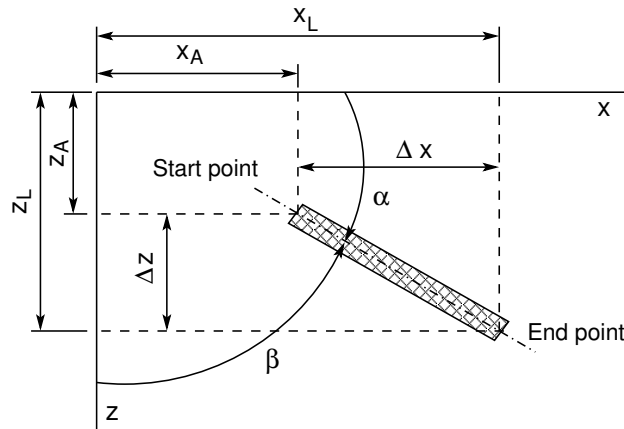


Figure A.4. Direction cosines of an element

$$\cos \alpha = \frac{x_L - x_A}{l} \quad (\text{A.21})$$

$$\cos \beta = \frac{z_L - z_A}{l} \quad (\text{A.22})$$

We now consider the transformation of the vector $\vec{\mathbf{F}}$ components F_x, F_z of Eq. (A.15) from the global xy coordinate system to the components F_x^*, F_z^* in the local x^*y^* coordinate system.

$$\begin{bmatrix} F_x^* \\ F_z^* \end{bmatrix} = \begin{bmatrix} \cos \alpha & \cos \beta \\ -\cos \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (\text{A.23})$$

Taking into account that $\cos \beta = \cos(90^\circ + \alpha) = -\sin \alpha$, we can write the above equation as

$$\begin{bmatrix} F_x^* \\ F_z^* \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} F_x \\ F_z \end{bmatrix} \quad (\text{A.24})$$

The inverse transformation of the vector $\vec{\mathbf{F}}$ components F_x^*, F_z^* of Eq. (A.15) from the local x^*y^* coordinate system to the components F_x, F_z of the global xy coordinate system:

$$\begin{bmatrix} F_x \\ F_z \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\cos \beta \\ \cos \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} F_x^* \\ F_z^* \end{bmatrix} \quad (\text{A.25})$$

Taking into account that $\cos \beta = \cos(90^\circ + \alpha) = -\sin \alpha$, we can write Eq. (A.25) in the form

$$\begin{bmatrix} F_x \\ F_z \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} F_x^* \\ F_z^* \end{bmatrix} \quad (\text{A.26})$$

Comparing the transformation matrix (A.23) with that of (A.25), we can see that they are transposed (rows and columns reversed). The multiplication of the matrix (A.23) by a transpose of itself of Eq. (A.25) gives the identity matrix

$$\begin{bmatrix} \cos \alpha & \cos \beta \\ -\cos \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \alpha & -\cos \beta \\ \cos \beta & \cos \alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{A.27})$$

We have thus proved that the matrices are *orthogonal* – an orthogonal matrix has the property that its transpose equals the inverse.

B. Work and work-energy theorem

B.1 Work done by internal and external forces

The work-energy theorem in structural analysis: the sum of the work done by internal and external forces is zero:

$$W_i + W_e = 0 \quad (\text{B.1})$$

where W_i is the work done by internal forces and W_e is the work done by external forces. The Green's functional for a frame element is [KHMW10]

$$\underbrace{- \int_a^b N_x \hat{\lambda} dx - \int_a^b M_y \hat{\psi}_y dx}_{W_i - \text{work of internal forces}} + \underbrace{[N_x \hat{u}]_a^b + [Q_y \hat{w} + M_y \hat{\varphi}_y]_a^b + \int_a^b q_x(x) \hat{u} dx + F_{xi} \hat{u}_i + \int_a^b q_z(x) \hat{w} dx + F_{zi} \hat{w}_i}_{W_e - \text{work of external forces}} = 0 \quad (\text{B.2})$$

where we consider two systems (states) of forces associated with respective deformations and displacements [BP13]:

N_x, M_y are the internal axial force and bending moment of the first load state;

$\hat{\lambda}, \hat{\psi}_y$ – axial and bending deformations of the second load state;

$N_x|_a^b, Q_y|_a^b, M_y|_a^b$ – axial force, shear force and bending moment of the first load state at boundaries a and b ;

$\hat{u}|_a^b, \hat{w}|_a^b, \hat{\varphi}_y|_a^b$ – longitudinal and transverse displacements, and the rotation of the cross section of the second load state at boundaries a and b ;

$q_x(x), q_z(x)$ – distributed loads of the first load state;

$\hat{u}(x), \hat{w}(x)$ – longitudinal and transverse displacements of the second load state;

F_{xi}, F_{zi} – force components of the first load state, applied at point i in the x and z directions, respectively;

\hat{u}_i, \hat{w}_i – longitudinal and transverse displacements of point i of the second load state.

The first and second elements of Eq. (B.2) describe the work W_i of internal forces:

$$W_i = - \int_a^b N_x \hat{\lambda} dx - \int_a^b M_y \hat{\psi}_y dx \quad (\text{B.3})$$

The final four elements of Eq. (B.2) describe the work W_f done by active forces:

$$W_f = \int_a^b q_x(x) \hat{u} dx + F_{xi} \hat{u}_i + \int_a^b q_z(x) \hat{w} dx + F_{zi} \hat{w}_i \quad (\text{B.4})$$

The third and fourth elements of Eq. (B.2) describe the work W_b done by boundary forces (fixed-end forces and moments at joints¹, support reactions):

$$W_b = [N_x \hat{u}]_a^b + [Q_y \hat{w} + M_y \hat{\varphi}_y]_a^b \quad (\text{B.5})$$

The external work W_e can be divided into two parts:

- W_f – work done by active forces, e.g. concentrated loads, uniformly distributed loads;
- W_b – work done by reaction forces, e.g. support reactions (Fig. 1.10), internal reactions [WP960] (contact forces²) (Fig. 1.12).

$$W_e = W_b + W_f \quad (\text{B.6})$$

Applying now Eq. (B.6) to (B.1), we obtain

$$W_i + W_b + W_f = 0 \quad (\text{B.7})$$

Equation (B.7) is a shortened form of Eq. (B.2).

With the elastic energy $U_{elastic\ energy}$ and dissipation energy D existing, the work W_i done by internal forces is in relation to the internal energy $U_{elastic} + D$:

$$\underbrace{W_i}_{\text{internal work}} = - \underbrace{U}_{\text{elastic energy}} - \underbrace{D}_{\text{dissipation energy}} \quad (\text{B.8})$$

Equations (B.2) and (B.7) are the basic methodical tools of structural analysis.

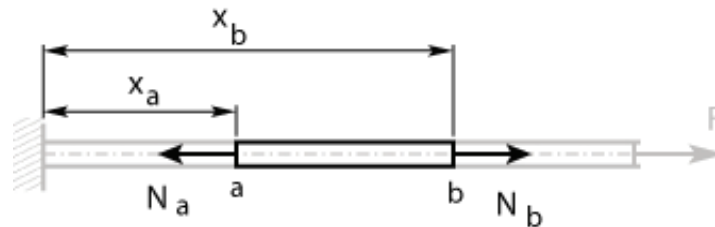


Figure B.1. Bar member a–b

¹The fixed-end forces and moments at joints are called the internal reactions [WP960] or the joint contact forces [GN12].

²A *contact force* is a force that acts at the points of contact between two objects [Rand07].

Example B.1 (conservation of mechanical energy). We consider the bar from Fig. B.1, subjected to the load F . The bar has been split into sections where the normal forces N_a and N_b exerted on the member at cross-sections a and b are treated as external loads [VrCty] or, to be more precise, as boundary forces.

No loads act on the bar member a - b , and so $W_f = 0$ and $W_i \neq 0$ in Eq. (B.7). The expression for energy conservation of the bar member a - b is

$$W_i + W_b = 0 \tag{B.9}$$

or

$$\underbrace{- \int_a^b N_x \hat{\lambda} dx}_{W_i - \text{work of internal forces}} + \underbrace{[N_x \hat{u}]_a^b}_{W_b - \text{work of internal reactions [WP960]}} = 0 \tag{B.10}$$

C. Transfer matrices

C.1 Transfer matrices of first-order analysis

Transfer matrices of first-order structural analysis are treated in [Bor79a], [Krä91a], and [PW94]. The transfer equation for a frame element is

$$\mathbf{Z}_p = \mathbf{U} \cdot \mathbf{Z}_v + \overset{\circ}{\mathbf{Z}} \quad (\text{C.1})$$

where

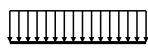
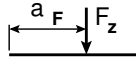
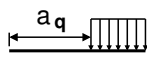
$$\mathbf{Z}_p = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_p, \quad \mathbf{Z}_v = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_v \quad (\text{C.2})$$

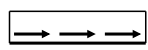
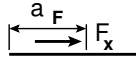
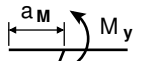
and \mathbf{U}_x is the transfer matrix (Sign Convention 2):

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 & -\frac{(x_p-x_v)}{EA} & 0 & 0 \\ 0 & 1 & -(x_p-x_v) & 0 & \frac{(x_p-x_v)^3}{6EI_y} - \frac{(x_p-x_v)}{GA_{red}} & \frac{(x_p-x_v)^2}{2EI_y} \\ 0 & 0 & 1 & 0 & -\frac{(x_p-x_v)^2}{2EI_y} & -\frac{(x_p-x_v)}{EI_y} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -(x_p-x_v) & -1 \end{bmatrix} \quad (\text{C.3})$$

$$\overset{\circ}{\mathbf{Z}} = \begin{bmatrix} \int_{x_v}^{x_p} \frac{1}{EA} \int_{x_v}^{x_p} q_x dx dx \\ \int_{x_v}^{x_p} \int_{x_v}^{x_p} \frac{1}{EI_y} \int_{x_v}^{x_p} \int_{x_v}^{x_p} q_z dx^2 dx^2 \\ - \int_{x_v}^{x_p} \frac{1}{EI_y} \int_{x_v}^{x_p} \int_{x_v}^{x_p} q_z dx^2 dx \\ - \int_{x_v}^{x_p} q_x dx dx \\ - \int_{x_v}^{x_p} q_z dx dx \\ - \int_{x_v}^{x_p} \int_{x_v}^{x_p} q_z dx dx + \int_{x_v}^{x_p} m_y dx \end{bmatrix} \quad (\text{C.4})$$

Table C.1. Loading vectors (to be continued in Table C.2)

$\overset{\circ}{\mathbf{Z}}$	 q_z		 q_z
$\overset{\circ}{u}$	0.0	0.0	0.0
$\overset{\circ}{w}$	$\frac{q_z \cdot x^4}{24EI_y}$	$\frac{F_z \cdot (x - a_F)_+^3}{6EI_y}$	$\frac{q_z \cdot (x - a_q)_+^4}{24EI_y}$
$\overset{\circ}{\varphi}_y$	$-\frac{q_z \cdot x^3}{6EI_y}$	$-\frac{F_z \cdot (x - a_F)_+^2}{2EI_y}$	$\frac{q_z \cdot (x - a_q)_+^3}{6EI_y}$
$\overset{\circ}{N}_x$	0.0	0.0	0.0
$\overset{\circ}{Q}_z$	$-q_z \cdot x$	$-F_z \cdot (x - a_F)_+^0$	$-q_z \cdot (x - a_q)_+^1$
$\overset{\circ}{M}_y$	$-\frac{q_z \cdot x^2}{2}$	$-F_z \cdot (x - a_F)_+$	$-\frac{q_z \cdot (x - a_q)_+^2}{2}$

$\overset{\circ}{\mathbf{Z}}$	 q_x		 M_y
$\overset{\circ}{u}$	$-\frac{q_x \cdot x^2}{2EA}$	$-\frac{q_x \cdot (x - a_F)_+}{EA}$	0.0
$\overset{\circ}{w}$	0.0	0.0	$\frac{M_y \cdot (x - a_M)_+^2}{2EI_y}$
$\overset{\circ}{\varphi}_y$	0.0	0.0	$-\frac{M_y \cdot (x - a_M)_+}{EI_y}$
$\overset{\circ}{N}_x$	$-q_x \cdot x$	$-F_x \cdot (x - a_F)_+^0$	0.0
$\overset{\circ}{Q}_z$	0.0	0.0	0.0
$\overset{\circ}{M}_y$	0.0	0.0	$-M_y \cdot (x - a_M)_+^0$

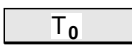

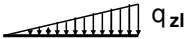
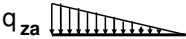
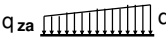
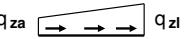
The loading vector $\overset{\circ}{\mathbf{Z}}_q$ for a uniformly distributed load q_z :

$$\overset{\circ}{\mathbf{Z}}_q = \begin{bmatrix} 0 \\ \frac{q_z \cdot (x_p - x_v)^4}{24EI_y} \\ -\frac{q_z \cdot (x_p - x_v)^3}{6EI_y} \\ 0 \\ -q_z \cdot (x_p - x_v) \\ -q_z \cdot (x_p - x_v)^2 \end{bmatrix} \quad (\text{C.5})$$

The loading vector $\overset{\circ}{\mathbf{Z}}_F$ for a point load F_z :

$$\overset{\circ}{\mathbf{Z}}_F = \begin{bmatrix} 0 \\ \frac{F_z \cdot (x_p - x_a)^3}{6EI_y} \\ -\frac{F_z \cdot (x_p - x_a)^2}{2EI_y} \\ 0 \\ -F_z \\ -F_z \cdot (x_p - x_a) \end{bmatrix} \quad (\text{C.6})$$

Table C.2. Loading vectors (continued from Table C.1)

$\overset{\circ}{\mathbf{Z}}$			
$\overset{\circ}{u}$	$\alpha_T T_0 x$	0.0	0.0
$\overset{\circ}{w}$	0.0	$-\alpha_T x^2 \frac{\Delta T}{2h}$	$\frac{q_{zl} \cdot l^4}{120EI_y}$
$\overset{\circ}{\varphi}_y$	0.0	$\alpha_T x \frac{\Delta T}{h}$	$-\frac{q_{zl} \cdot l^3}{24EI_y}$
$\overset{\circ}{N}_x$	0.0	0.0	0.0
$\overset{\circ}{Q}_z$	0.0	0.0	$-\frac{q_z \cdot l}{2}$
$\overset{\circ}{M}_y$	0.0	0.0	$-\frac{q_z \cdot l^2}{6}$
$\overset{\circ}{\mathbf{Z}}$			
$\overset{\circ}{u}$	0.0	0.0	$-\frac{l^2}{6EA} (2q_{xa} + q_{xl})$
$\overset{\circ}{w}$	$\frac{q_{za} \cdot l^4}{30EI_y}$	$\frac{(4q_{za} + q_{zl})l^4}{120EI_y}$	0.0
$\overset{\circ}{\varphi}_y$	$-\frac{q_{za} \cdot l^3}{8EI_y}$	$-\frac{(3q_{za} + q_{zl})l^3}{24EI_y}$	0.0
$\overset{\circ}{N}_x$	0.0	0.0	$-\frac{1}{2} (q_{xa} + q_{xl})$
$\overset{\circ}{Q}_z$	$-\frac{q_{za} \cdot l}{2}$	$-\frac{1}{2} (q_{za} + q_{zl})$	0.0
$\overset{\circ}{M}_y$	$-\frac{q_{za} \cdot l^2}{3}$	$-\frac{l^2}{2} (2q_{za} + q_{zl})$	0.0

The transfer matrix \mathbf{U} of the first-order analysis for a frame (Sign Convention 2) can be computed using the GNU Octave function `ylfhlin.m` (p. 242); for a sparse transfer matrix \mathbf{U} , the function `ysplfhlin.m` (p. 244) is to be used.

To compute the enlarged sparse transfer matrix $\widehat{\mathbf{U}}$ of the first-order analysis

- for a beam (Sign Convention 2), the GNU Octave function `yspTlvfmhvI.m` (p. 244) is used. This function in its turn makes use of the function `yspTlfhlin.m` (p. 244);
- for a three-hinged frame (Sign Convention 2), the GNU Octave function `yspSlvfmhvI.m` (p. 245), which comprises the function `yspSlfhlin.m` (p. 244), is used;
- for a Gerber beam (Sign Convention 2), the GNU Octave function `yspSTlvfmhvI.m` (p. 245) comprising the function `yspSTlfhlin.m` (p. 244) is used.

The loading vectors in Tables C.1 and C.2 are comparable with those in [Krä91b] and [Bor79b].

The loading vectors $\overset{\circ}{\mathbf{Z}}_{\mathbf{q}}$, $\overset{\circ}{\mathbf{Z}}_{\mathbf{F}}$ can be computed using the GNU Octave functions `yzhqz.m` (p. 242) and `yzfzv.m` (p. 243).

For a continuous beam, the loading vector $\overset{\circ}{\mathbf{Z}}$ can be computed using the GNU Octave function `ESTtalaKrmus.m` (p. 245); to compute the vectors $\overset{\circ}{\mathbf{Z}}_{\mathbf{q}}$, $\overset{\circ}{\mathbf{Z}}_{\mathbf{F}}$, the functions `yzThqz.m` and `yzTfzv.m` (p. 243) are used.

For a three-hinged frame, the loading vector $\overset{\circ}{\mathbf{Z}}$ can be computed using the GNU Octave function `ESTSKrmus.m` (p. 245); to compute the vectors $\overset{\circ}{\mathbf{Z}}_{\mathbf{q}}$, $\overset{\circ}{\mathbf{Z}}_{\mathbf{F}}$, the functions `yzShqz.m` and `yzSfzv.m` (p. 243) are used.

For a Gerber beam, the loading vector $\overset{\circ}{\mathbf{Z}}$ can be computed using the GNU Octave function `ESTSTKrmus.m` (p. 246); to compute the vectors $\overset{\circ}{\mathbf{Z}}_{\mathbf{q}}$, $\overset{\circ}{\mathbf{Z}}_{\mathbf{F}}$, the functions `yzSThqz.m` and `yzSTfzv.m` (p. 243) are used.

C.2 Transfer matrices of second-order analysis

The transfer equations with axial and transverse shear forces are of the form

$$\tilde{\mathbf{Z}}_{\mathbf{p}} - \tilde{\mathbf{U}}\tilde{\mathbf{Z}}_{\mathbf{v}} = \overset{\circ}{\mathbf{Z}} \quad (\text{C.7})$$

where

$$\tilde{\mathbf{Z}}_{\mathbf{p}} = \begin{bmatrix} u \\ w \\ \varphi_y \\ S_x \\ H_z \\ M_y \end{bmatrix}_p, \quad \tilde{\mathbf{Z}}_{\mathbf{v}} = \begin{bmatrix} u \\ w \\ \varphi_y \\ S_x \\ H_z \\ M_y \end{bmatrix}_v \quad (\text{C.8})$$

where

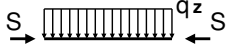
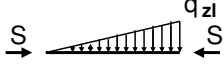
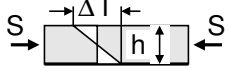
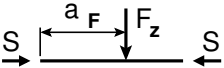
S – axial force,

H – transverse shear force, see Fig. 5.1.

Numerical difficulties may occur when the transfer matrix manipulation involves differences of large numbers (forces and displacements can differ ca 10^3 times), leading to inaccuracies in computations [PW94], [PL63]. Round-off errors can be reduced by scaling. We will scale (multiply) the displacements and rotations by $i_0 = EI_{basic}/l_{basic}$ (the scaling multiplier for the displacements). After solving a system of linear equations of a boundary value problem, we divide each of the displacements and rotations found by $i_0 = EI_{basic}/l_{basic}$. Displacements and forces at the beginning of members determine unscaled initial parameter vectors for the structure members.

The transfer matrix \mathbf{U} from Eqs. (C.7), (C.9), and (C.10) of a second-order analysis for a frame (Sign Convention 2) can be computed using the GNU Octave function `ylfmhvII.m` (p. 248).

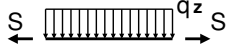
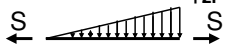
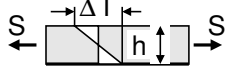
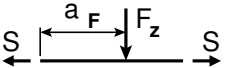
Table C.3. Loading vectors of compression (axial force S and transverse shear force H)

$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{ql^4}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \cos \frac{\nu x}{l} \right]$	$-\frac{ql^4}{\nu^5 EI} \left[\frac{\nu x}{l} - \frac{1}{6} \left(\frac{\nu x}{l} \right)^3 - \sin \frac{\nu x}{l} \right]$
$\overset{\circ}{\varphi}_y$	$-\frac{ql^3}{\nu^3 EI} \left[\frac{\nu x}{l} - \sin \frac{\nu x}{l} \right]$	$-\frac{ql^3}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \cos \frac{\nu x}{l} \right]$
$\overset{\circ}{S}_x$	0.0	0.0
$\overset{\circ}{H}_z$	$-\frac{ql}{\nu} \sin \frac{\nu x}{l}$	$-\frac{ql}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$
$\overset{\circ}{M}_y$	$-\frac{ql^2}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$	$-\frac{ql^2}{\nu^3} \left[\frac{\nu x}{l} - \sin \frac{\nu x}{l} \right]$
$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{\alpha_t \Delta T}{h} \frac{l^2}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$	$\left[\frac{\nu}{l} (x - a) - \right] \frac{Fl^3}{EI\nu^3}$
$\overset{\circ}{\varphi}_y$	$-\frac{\alpha_T \Delta T}{h} \frac{l^2}{\nu^2} \sin \frac{\nu x}{l}$	$-\left[\cos \frac{\nu}{l} (x - a) - 1 \right] \frac{Fl^2}{EI\nu 12}$
$\overset{\circ}{S}_x$	0.0	0.0
$\overset{\circ}{H}_z$	$-\frac{\alpha_T \Delta T}{h} EI \frac{\nu}{l} \sin \frac{\nu x}{l}$	$-F$
$\overset{\circ}{M}_y$	$-\frac{\alpha_T \Delta T}{h} EI \left[1 - \cos \frac{\nu x}{l} \right]$	$-\left[\sin \frac{\nu}{l} (x - a) \right] \frac{Fl}{\nu}$

The transfer matrix \mathbf{U} for a frame in compression, Sign Convention 2:

$$\tilde{\mathbf{U}} = \begin{bmatrix} 1 & 0 & 0 & -i_o \frac{x}{EA} & 0 & 0 \\ 0 & 1 & -\frac{\sin \nu \frac{\nu x}{l}}{\nu} x & 0 & i_o \left(\frac{\frac{\nu x}{l} - \sin \frac{\nu x}{l}}{\nu^3} \frac{l^3}{EI} \right) & i_o \left(\frac{1 - \cos \frac{\nu x}{l}}{\nu^2} \frac{l^2}{EI} \right) \\ 0 & 0 & \cos \frac{\nu x}{l} & 0 & -i_o \left(\frac{-1 + \cos \frac{\nu x}{l}}{\nu^2} \frac{l^2}{EI} \right) & -i_o \left(\frac{-\sin \frac{\nu x}{l}}{\nu} \frac{l}{EI} \right) \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -\nu \sin \frac{\nu x}{l} \frac{l}{EI} i_o & 0 & -\frac{\sin \frac{\nu x}{l}}{\nu} x & -\cos \frac{\nu x}{l} \end{bmatrix} \quad (\text{C.9})$$

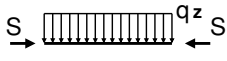
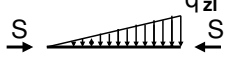
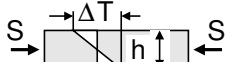
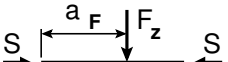
Table C.4. Loading vectors of tension (axial force S and transverse shear force H)

$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{ql^4}{\nu^4 EI} \left[1 + \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \text{ch} \frac{\nu x}{l} \right]$	$-\frac{ql^4}{\nu^5 EI} \left[\frac{\nu x}{l} - \frac{1}{6} \left(\frac{\nu x}{l} \right)^3 - \text{sh} \frac{\nu x}{l} \right]$
$\overset{\circ}{\varphi}_y$	$-\frac{ql^3}{\nu^3 EI} \left[\frac{\nu x}{l} - \text{sh} \frac{\nu x}{l} \right]$	$-\frac{ql^3}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \text{ch} \frac{\nu x}{l} \right]$
$\overset{\circ}{S}_x$	0.0	0.0
$\overset{\circ}{H}_z$	$-\frac{ql}{\nu} \text{sh} \frac{\nu x}{l}$	$-\frac{ql}{\nu^2} \left[1 - \text{ch} \frac{\nu x}{l} \right]$
$\overset{\circ}{M}_y$	$-\frac{ql^2}{\nu^2} \left[1 - \text{ch} \frac{\nu x}{l} \right]$	$-\frac{ql^2}{\nu^3} \left[\frac{\nu x}{l} - \text{sh} \frac{\nu x}{l} \right]$
$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{\alpha_t \Delta T}{h} \frac{l^2}{\nu^2} \left[1 - \text{ch} \frac{\nu x}{l} \right]$	$\left[\frac{\nu}{l} (x - a) - \text{sh} \frac{\nu}{l} (x - a) \right] \frac{Fl^3}{EI\nu^3}$
$\overset{\circ}{\varphi}_y$	$-\frac{\alpha_T \Delta T}{h} \frac{l^2}{\nu^2} \text{sh} \frac{\nu x}{l}$	$-\left[\text{ch} \frac{\nu}{l} (x - a) - 1 \right] \frac{Fl^2}{EI\nu^2}$
$\overset{\circ}{S}_x$	0.0	0.0
$\overset{\circ}{H}_z$	$-\frac{\alpha_T \Delta T}{h} EI \frac{\nu}{l} \text{sh} \frac{\nu x}{l}$	$-F$
$\overset{\circ}{M}_y$	$-\frac{\alpha_T \Delta T}{h} EI \left[1 - \text{ch} \frac{\nu x}{l} \right]$	$-\left[\text{sh} \frac{\nu}{l} (x - a) \right] \frac{Fl}{\nu}$

The transfer matrix \mathbf{U} for a frame in tension, Sign Convention 2:

$$\tilde{\mathbf{U}}_t = \begin{bmatrix} 1 & 0 & 0 & -i_o \frac{x}{EA} & 0 & 0 \\ 0 & 1 & -\frac{\text{sh} \nu \frac{\nu x}{l}}{\nu} x & 0 & i_o \left(\frac{-\nu x}{l} + \text{sh} \frac{\nu x}{l} \frac{l^3}{EI} \right) & i_o \left(\frac{-1 - \text{ch} \frac{\nu x}{l}}{\nu^2} \frac{l^2}{EI} \right) \\ 0 & 0 & \text{ch} \frac{\nu x}{l} & 0 & i_o \left(\frac{1 - \text{ch} \frac{\nu x}{l}}{\nu^2} \frac{l^2}{EI} \right) & i_o \left(\frac{-\text{sh} \frac{\nu x}{l}}{\nu} \frac{l}{EI} \right) \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & \nu \text{sh} \frac{\nu x}{l} \frac{l}{EI} \frac{1}{i_o} & 0 & \frac{-\text{sh} \frac{\nu x}{l}}{\nu} x & -\text{ch} \frac{\nu x}{l} \end{bmatrix} \quad (\text{C.10})$$

Table C.5. Loading vectors of compression (normal force N and shear force Q)

$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{ql^4}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \cos \frac{\nu x}{l} \right]$	$-\frac{ql^4}{\nu^5 EI} \left[\frac{\nu x}{l} - \frac{1}{6} \left(\frac{\nu x}{l} \right)^3 - \sin \frac{\nu x}{l} \right]$
$\overset{\circ}{\varphi}_y$	$-\frac{ql^3}{\nu^3 EI} \left[\frac{\nu x}{l} - \sin \frac{\nu x}{l} \right]$	$-\frac{ql^3}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - \cos \frac{\nu x}{l} \right]$
$\overset{\circ}{N}_x$	0.0	0.0
$\overset{\circ}{Q}_z$	$-\frac{ql}{\nu} \sin \frac{\nu x}{l}$	$-\frac{ql}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$
$\overset{\circ}{M}_y$	$-\frac{ql^2}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$	$-\frac{ql^2}{\nu^3} \left[\frac{\nu x}{l} - \sin \frac{\nu x}{l} \right]$
$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{\alpha_t \Delta t}{h} \frac{l^2}{\nu^2} \left[1 - \cos \frac{\nu x}{l} \right]$	$\left[\frac{\nu}{l} (x - a) - \right] \frac{Fl^3}{EI\nu^3}$
$\overset{\circ}{\varphi}_y$	$-\frac{\alpha_T \Delta T}{h} \frac{l^2}{\nu^2} \sin \frac{\nu x}{l}$	$\left[\cos \frac{\nu}{l} (x - a) - 1 \right] \frac{Fl^2}{EI\nu 12}$
$\overset{\circ}{N}_x$	0.0	0.0
$\overset{\circ}{Q}_z$	$-\frac{\alpha_T \Delta T}{h} EI \frac{\nu}{l} \sin \frac{\nu x}{l}$	$-F$
$\overset{\circ}{M}_y$	$-\frac{\alpha_T \Delta T}{h} EI \left[1 - \cos \frac{\nu x}{l} \right]$	$-\left[\sin \frac{\nu}{l} (x - a) \right] \frac{Fl}{\nu}$

In the case of a uniformly distributed load q_z , the loading vector $\overset{\circ}{\mathbf{Z}}$ (see Eq. (C.7)) of a second-order analysis for a frame member under axial compression S_x is

$$\overset{\circ}{\mathbf{Z}} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ i_o \left[\frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - 1 + \cos \left(\frac{\nu x}{l} \right) \right] \frac{ql^4}{\nu^4 EI} \\ -i_o \left[\frac{\nu x}{l} - \sin \left(\frac{\nu x}{l} \right) \right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ -q \cdot x \\ - \left[1 - \cos \left(\frac{\nu x}{l} \right) \right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (\text{C.11})$$

Table C.6. Loading vectors of tension (normal force N and shear force Q)

$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{ql^4}{\nu^4 EI} \left[1 + \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - ch \frac{\nu x}{l} \right]$	$-\frac{ql^4}{\nu^5 EI} \left[\frac{\nu x}{l} - \frac{1}{6} \left(\frac{\nu x}{l} \right)^3 - sh \frac{\nu x}{l} \right]$
$\overset{\circ}{\varphi}_y$	$-\frac{ql^3}{\nu^3 EI} \left[\frac{\nu x}{l} - sh \frac{\nu x}{l} \right]$	$-\frac{ql^3}{\nu^4 EI} \left[1 - \frac{1}{2} \left(\frac{\nu x}{l} \right)^2 - ch \frac{\nu x}{l} \right]$
$\overset{\circ}{N}_x$	0.0	0.0
$\overset{\circ}{Q}_z$	$-\frac{ql}{\nu} sh \frac{\nu x}{l}$	$-\frac{ql}{\nu^2} \left[1 - ch \frac{\nu x}{l} \right]$
$\overset{\circ}{M}_y$	$-\frac{ql^2}{\nu^2} \left[1 - ch \frac{\nu x}{l} \right]$	$-\frac{ql^2}{\nu^3} \left[\frac{\nu x}{l} - sh \frac{\nu x}{l} \right]$
$\overset{\circ}{\mathbf{Z}}$		
$\overset{\circ}{u}$	0.0	0.0
$\overset{\circ}{w}$	$-\frac{\alpha_t \Delta t}{h} \frac{l^2}{\nu^2} \left[1 - ch \frac{\nu x}{l} \right]$	$\left[\frac{\nu}{l} (x - a) - sh \frac{\nu}{l} (x - a) \right] \frac{Fl^3}{EI\nu^3}$
$\overset{\circ}{\varphi}_y$	$-\frac{\alpha_T \Delta T}{h} \frac{l^2}{\nu^2} sh \frac{\nu x}{l}$	$-\left[ch \frac{\nu}{l} (x - a) - 1 \right] \frac{Fl^2}{EI\nu^2}$
$\overset{\circ}{N}_x$	0.0	0.0
$\overset{\circ}{Q}_z$	$-\frac{\alpha_T \Delta T}{h} EI \frac{\nu}{l} sh \frac{\nu x}{l}$	$-F$
$\overset{\circ}{M}_y$	$-\frac{\alpha_T \Delta T}{h} EI \left[1 - ch \frac{\nu x}{l} \right]$	$-\left[sh \frac{\nu}{l} (x - a) \right] \frac{Fl}{\nu}$

and for a frame member under axial tension S_x ,

$$\overset{\circ}{\mathbf{Z}}_{\mathbf{t}} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ -i_o \left[\frac{1}{2} \left(\nu \frac{x}{l} \right)^2 - 1 - ch \left(\nu \frac{x}{l} \right) \right] \frac{ql^4}{\nu^4 EI} \\ i_o \left[\nu \frac{x}{l} - sh \left(\nu \frac{x}{l} \right) \right] \frac{ql^3}{\nu^3 EI} \\ 0 \\ -q \cdot x \\ \left[1 - ch \left(\nu \frac{x}{l} \right) \right] \frac{ql^2}{\nu^2} \end{bmatrix} \quad (\text{C.12})$$

The loading vectors in Tables C.3, C.4, C.5, and C.6 are comparable with those in [Krä91b] and [Bor79b].

The loading vectors of Eqs. (C.11) and (C.12) can be computed using the GNU Octave function `ylqvII.m` (p. 249).

In the case of a point load F_z , the loading vector $\overset{\circ}{\mathbf{Z}}$ (see Eq. (C.7)) of a second-order analysis for a frame member under axial compression S_x is

$$\overset{\circ}{\mathbf{Z}} = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ -i_o \left[\sin \left(\nu \frac{(x-a)_+}{l} \right) - \left(\nu \frac{(x-a)_+}{l} \right) \right] \frac{Fl^3}{\nu^3 EI} \\ i_o \left[\cos \left(\nu \frac{(x-a)_+}{l} \right) - 1 \right] \frac{Fl^2}{\nu^2 EI} \\ 0 \\ -(x-a)_+^0 F \\ - \left[\sin \left(\nu \frac{(x-a)_+}{l} \right) \right] \frac{Fl}{\nu} \end{bmatrix} \quad (\text{C.13})$$

and for a frame member under axial tension S_x ,

$$\overset{\circ}{\mathbf{Z}}_t = \begin{bmatrix} u_e \\ w_e \\ \varphi_e \\ S_e \\ H_e \\ M_e \end{bmatrix} = \begin{bmatrix} 0 \\ i_o \left[\text{sh} \left(\nu \frac{(x-a)_+}{l} \right) - \left(\nu \frac{(x-a)_+}{l} \right) \right] \frac{Fl^3}{\nu^3 EI} \\ -i_o \left[\text{ch} \left(\nu \frac{(x-a)_+}{l} \right) - 1 \right] \frac{Fl^2}{\nu^2 EI} \\ 0 \\ -(x-a)_+^0 F \\ - \left[\text{sh} \left(\nu \frac{(x-a)_+}{l} \right) \right] \frac{Fl}{\nu} \end{bmatrix} \quad (\text{C.14})$$

where $(x-a)_+$ is the Heaviside function:

$$(x-a)_+ = \begin{cases} 0, & \text{if } (x-a) < 0 \\ x-a, & \text{if } (x-a) \geq 0 \end{cases} \quad (\text{C.15})$$

The loading vectors of Eqs. (C.13) and (C.14) can be computed using the GNU Octave function `ylfhvzII.m` (p. 248).

The basic system of equations of the frame element of Eq. (C.7) can be rewritten in the form

$$\widehat{\mathbf{U}} \cdot \widehat{\mathbf{Z}} = \overset{\circ}{\mathbf{Z}} \quad (\text{C.16})$$

where the matrix $\widehat{\mathbf{U}}$ can be computed using the GNU Octave function `ysplvfmhvII.m` (p. 250), in which the GNU Octave function `ysplfmhvII.m` (p. 248) is used. These functions differentiate between pressure and tensile axial loading.

The loading vectors of a uniformly distributed load and a point load for a frame can be computed using the GNU Octave functions `ylqvII.m` (p. 249) and `ylfhvzII.m` (p. 248). These functions differentiate between pressure and tensile axial loading.

The transfer equations with normal and shear forces (N – normal force, Q – shear force shown in Fig. 5.1):

$$\mathbf{Z}_p = \mathbf{U}_x \cdot \mathbf{Z}_v + \overset{\circ}{\mathbf{Z}} \quad (\text{C.17})$$

where

$$\mathbf{Z}_p = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_p, \quad \mathbf{Z}_v = \begin{bmatrix} u \\ w \\ \varphi_y \\ N_x \\ Q_z \\ M_y \end{bmatrix}_v \quad (\text{C.18})$$

and the transfer matrix $\mathbf{U}_x \equiv \mathbf{U}^{(\mathbf{N}-2)}$ for a compressive normal force (Sign Convention 2)

$$\mathbf{U}^{(\mathbf{N}-2)} = \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & \left[\nu \frac{x}{l} - \sin\left(\nu \frac{x}{l}\right) \right] \frac{l^3}{\nu^3 EI_y} & \left[1 - \cos\left(\nu \frac{x}{l}\right) \right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & -\left[1 - \cos\left(\nu \frac{x}{l}\right) \right] \frac{l^2}{\nu^2 EI_y} & -\sin\left(\nu \frac{x}{l}\right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\cos\left(\nu \frac{x}{l}\right) & \sin\left(\nu \frac{x}{l}\right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & -\sin\left(\nu \frac{x}{l}\right) \frac{l}{\nu} & -\cos\left(\nu \frac{x}{l}\right) \end{bmatrix} \quad (\text{C.19})$$

while $\mathbf{U}_x \equiv \mathbf{U}^{(\mathbf{N}+2)}$ for a tensile normal force (Sign Convention 2)

$$\mathbf{U}^{(\mathbf{N}+2)} = \begin{bmatrix} 1 & 0 & 0 & -\frac{x}{EA} & 0 & 0 \\ 0 & 1 & -x & 0 & -\left[\nu \frac{x}{l} - \text{sh}\left(\nu \frac{x}{l}\right) \right] \frac{l^3}{\nu^3 EI_y} & -\left[1 - \text{ch}\left(\nu \frac{x}{l}\right) \right] \frac{l^2}{\nu^2 EI_y} \\ 0 & 0 & 1 & 0 & \left[1 - \text{ch}\left(\nu \frac{x}{l}\right) \right] \frac{l^2}{\nu^2 EI_y} & -\text{sh}\left(\nu \frac{x}{l}\right) \frac{l}{\nu EI_y} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\text{ch}\left(\nu \frac{x}{l}\right) & -\text{sh}\left(\nu \frac{x}{l}\right) \frac{\nu}{l} \\ 0 & 0 & 0 & 0 & -\text{sh}\left(\nu \frac{x}{l}\right) \frac{l}{\nu} & -\text{ch}\left(\nu \frac{x}{l}\right) \end{bmatrix} \quad (\text{C.20})$$

Transfer matrices (C.19) and (C.20) can be computed with the GNU Octave function [ylfmII.m](#) (p. 249).

D. Computer programs for the EST method

Computer programs for the EST method can be found on a CD attached to the book. The basic URLs (folder paths for the programs): D:/, E:/, F:/, S:/, Z:/.

D.1 Programs for first-order analysis

Program D.1 (spESTframeLaheWFI.m)¹ 58 – is used to compute the displacements and internal forces of a plane frame.

Called functions:

- *LaheFrameDFIm.m*;
- *SisejoudPunktism.m*.

Function D.1 (LaheFrameDFIm(baasi0,Ntoerkts,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem))² 34, 60, 68, 76, 235, 236, 237 – is used to assemble and solve the boundary problem equations of a plane frame.

Called functions:

- *yzhqzm(baasi0,x,a,qx,qz,EA,EJ)*³;
- *InsertBtoA(A,I,J,IM,JN,B,M,N)*⁴;
- *spInsertBtoA(spA,IJv,IJv,spvF)*⁵;
- *spSisestaArv(spA,iv,jv,sv)*⁶;
- *SpTeisendusMaatriks2x2(NSARV,NEARV,VarrasN,krdn,selem)*⁷;
- *SpTeisendusMaatriks(NSARV,NEARV,VarrasN,krdn,selem)*⁸;
- *SpTeisendusUhikMaatriks2x2(VarrasN)*⁹;
- *SpTeisendusUhikMaatriks0x1v(VarrasN)*¹⁰;

¹ ./octavePrograms/spESTframeLaheWFI.m

² ./octavePrograms/LaheFrameDFIm.m

³ ./octavePrograms/yzhqzm.m

⁴ ./octavePrograms/InsertBtoA.m

⁵ ./octavePrograms/spInsertBtoA.m

⁶ ./octavePrograms/spSisestaArv.m

⁷ ./octavePrograms/SpTeisendusMaatriks2x2.m

⁸ ./octavePrograms/SpTeisendusMaatriks.m

⁹ ./octavePrograms/SpTeisendusUhikMaatriks2x2.m

¹⁰ ./octavePrograms/SpTeisendusUhikMaatriks0x1v.m

- *SpTeisendusUhikMaatriks(VarrasN)* ¹¹;
- *SpToeReaktsioonZvektor(NSARV,NEARV,VarrasN,krdn,selem)* ¹²;
- *SpToeReaktsioonXvektor(NSARV,NEARV,VarrasN,krdn,selem)* ¹³;
- *SpToeSiirdeFiVektor(VarrasN)* ¹⁴;
- *SpToeSiirdeUvektor(NSARV,NEARV,VarrasN,krdn,selem)* ¹⁵;
- *SpToeSiirdeWvektor(NSARV,NEARV,VarrasN,krdn,selem)* ¹⁶;
- *VardadSolmes(NSARV,NEARV,Solm,AB,ABB)* ¹⁷;
- *VardaPikkus(NSARV,NEARV,krdn,selem)* ¹⁸;
- *ylfhlin(baasi0,x,EA,GAr,EJ)* ¹⁹;
- *ysplfhlin(baasi0,x,EA,GAr,EJ)* ²⁰;
- *ysplvmhvI(baasi0,x,l,EA,GAr,EJ)* ²¹;
- *yzfzv(baasi0,x,a,Fx,Fz,EA,EJ)* ²²;
- *yzhqzm(baasi0,x,a,qx,qz,EA,EJ)* ²³.

Function D.2 (*SisejoudPunktism(VardaNr,X,AlgPar,lvarras,selem,esQkoormus,esFjoud,suurused)*) ²⁴ 235, 236, 237 – is used to compute the displacements and internal forces of the element 'VardaNr' at $x = X$.

Called function:

- *ESTFrKrmus(baasi0,xx,Li,Fjoud,qkoormus,EA,EI)* ²⁵.

Program D.2 (*spESTframe93LaheWFI.m*) ²⁶ 53, 66 – is used to compute the displacements and internal forces of a plane frame.

Called functions:

- *LaheFrameDFIm.m*;
- *SisejoudPunktism.m*.

Program D.3 (*spESTframe77LaheWFI.m*) ²⁷ 74 – is used to compute the displacements and internal forces of a plane frame.

Called functions:

-
- ¹¹ `./octavePrograms/SpTeisendusUhikMaatriks.m`
 - ¹² `./octavePrograms/SpToeReaktsioonZvektor.m`
 - ¹³ `./octavePrograms/SpToeReaktsioonXvektor.m`
 - ¹⁴ `./octavePrograms/SpToeSiirdeFiVektor.m`
 - ¹⁵ `./octavePrograms/SpToeSiirdeUvektor.m`
 - ¹⁶ `./octavePrograms/SpToeSiirdeWvektor.m`
 - ¹⁷ `./octavePrograms/VardadSolmes.m`
 - ¹⁸ `./octavePrograms/VardaPikkus.m`
 - ¹⁹ `./octavePrograms/ylfhlin.m`
 - ²⁰ `./octavePrograms/ysplfhlin.m`
 - ²¹ `./octavePrograms/ysplvmhvI.m`
 - ²² `./octavePrograms/yzfzv.m`
 - ²³ `./octavePrograms/yzhqzm.m`
 - ²⁴ `./octavePrograms/SisejoudPunktism.m`
 - ²⁵ `./octavePrograms/ESTFrKrmus.m`
 - ²⁶ `./octavePrograms/spESTframe93LaheWFI.m`
 - ²⁷ `./octavePrograms/spESTframe77LaheWFI.m`

- *LaheFrameDFIm.m*;
- *SisejoudPunktism.m*.

Program D.4 (*spESTframe3hingeLaheWFI.m*)²⁸ 108 – is used to compute the displacements and internal forces of a plane frame.

Called functions:

- *LaheFrameDFIm.m*;
- *SisejoudPunktism.m*.

Program D.5 (*spESTframe3hingeLaheNQM.m*)²⁹ 104 – is used to compute the internal forces of a plane frame.

Called functions:

- *LaheFrame3hingeNQM.m*;
- *Sisejoud3LraamiPnktism.m*.

Function D.3 (*LaheFrame3hingeNQM(Ntoerkts,esQkoormus,esFjoud,sSolmF,tsolm,krdn,selem)*)³⁰ 106, 237 – is used to assemble and solve the boundary problem equations of a statically determinate plane frame.

Called functions:

- *ESTSKrmus(xx,Li,Fjoud,qkoormus)*³¹;
- *InsertBtoA(A,I,J,IM,JN,B,M,N)*³²;
- *spInsertBtoA(spA,IIV,IJv,spvF)*³³;
- *spSisestaArv(spA,iv,jv,sv)*³⁴;
- *SpTeisendusMaatriks2x2D(NSARV,NEARV,VarrasN,krdn,selem)*³⁵;
- *SpTeisendusMaatriksD(NSARV,NEARV,VarrasN,krdn,selem)*³⁶;
- *SpTeisendusUhikMaatriks0x1v(VarrasN)*³⁷;
- *SpTeisendusUhikMaatriks2x2(VarrasN)*³⁸;
- *SpTeisendusUhikMaatriks(VarrasN)*³⁹;
- *VardadSolmesD(NSARV,NEARV,Solm,AB,ABB)*⁴⁰;
- *VardaPikkusD(NSARV,NEARV,krdn,selem)*⁴¹;
- *ylSfhlin(x)*⁴²;

²⁸ `./octavePrograms/spESTframe3hingeLaheWFI.m`

²⁹ `./octavePrograms/spESTframe3hingeLaheNQM.m`

³⁰ `./octavePrograms/LaheFrame3hingeNQM.m`

³¹ `./octavePrograms/ESTSKrmus.m`

³² `./octavePrograms/InsertBtoA.m`

³³ `./octavePrograms/spInsertBtoA.m`

³⁴ `./octavePrograms/spSisestaArv.m`

³⁵ `./octavePrograms/SpTeisendusMaatriks2x2D.m`

³⁶ `./octavePrograms/SpTeisendusMaatriksD.m`

³⁷ `./octavePrograms/SpTeisendusUhikMaatriks0x1v.m`

³⁸ `./octavePrograms/SpTeisendusUhikMaatriks2x2.m`

³⁹ `./octavePrograms/SpTeisendusUhikMaatriks.m`

⁴⁰ `./octavePrograms/VardadSolmesD.m`

⁴¹ `./octavePrograms/VardaPikkusD.m`

⁴² `./octavePrograms/ylSfhlin.m`

- *ySpSlfhlin*(x)⁴³;
- *ySpSlvfmhvI*(x)⁴⁴;
- *yzSfzv*(x, a, Fx, Fz)⁴⁵;
- *yzShqz*(x, qx, qz)⁴⁶.

Function D.4 (*Sisejoud3LraamiPnktism*(*VardaNr*,*X*,*AlgPar*,*lvarras*,*esFjoud*,*esQkoormus*,*suurused*))⁴⁷ 237 – is used to compute the displacements and internal forces of the element 'VardaNr' at $x = X$.

Called function:

- *ESTSKrmus*($xx, Li, Fjoud, qkoormus$)⁴⁸.

Program D.6 (*spESTGerberBeamQM.m*)⁴⁹ 110 – is used to compute the internal forces of a Gerber beam.

Called functions:

- *LaheGerberBeamQM.m*;
- *SsjoudGrbrTalaPnktis.m*.

Function D.5 (*LaheGerberBeamQM*(*Ntoerks*,*esQkoormus*,*esFjoud*,*sSolmF*,*tsolm*,*krdn*,*selem*))⁵⁰ 112, 238 – is used to assemble and solve the boundary problem equations of a statically determinate beam.

Called functions:

- *VardaPikkusDT*(*NSARV*,*NEARV*,*krdn*,*selem*)⁵¹;
- *ySpSTlvfmhvI*(x)⁵²;
- *ySpSTlfhlin*(x)⁵³;
- *ESTSTKrmus*($xx, Li, Fjoud, qkoormus$)⁵⁴;
- *yzSTfzv*(x, a, Fz)⁵⁵;
- *yzSThqz*(x, qz)⁵⁶;
- *VardadSolmesDT*(*NSARV*,*NEARV*,*Solm*,*AB*,*ABB*)⁵⁷;
- *SpTeisendusUhhkMaatriks2x2*(*VarrasN*)⁵⁸;

⁴³ ./octavePrograms/ySpSlfhlin.m

⁴⁴ ./octavePrograms/ySpSlvfmhvI.m

⁴⁵ ./octavePrograms/yzSfzv.m

⁴⁶ ./octavePrograms/yzShqz.m

⁴⁷ ./octavePrograms/Sisejoud3LraamiPnktism.m

⁴⁸ ./octavePrograms/ESTFrKrmus.m

⁴⁹ ./octavePrograms/spESTGerberBeamQM.m

⁵⁰ ./octavePrograms/LaheGerberBeamQM.m

⁵¹ ./octavePrograms/VardaPikkusDT.m

⁵² ./octavePrograms/ySpSTlvfmhvI.m

⁵³ ./octavePrograms/ySpSTlfhlin.m

⁵⁴ ./octavePrograms/ESTSTKrmus.m

⁵⁵ ./octavePrograms/yzSTfzv.m

⁵⁶ ./octavePrograms/yzSThqz.m

⁵⁷ ./octavePrograms/VardadSolmesDT.m

⁵⁸ ./octavePrograms/SpTeisendusUhhkMaatriks2x2.m

- *SpTeisendusUhhkMaatriks1x0(VarrasN)*⁵⁹;
- *ylSTfhlin(x)*⁶⁰;
- *SsjoudGrbrTalaPunktis(VardaNr,X,AlgPar,lvarras,esFjoud,esQkoormus;suurused)*⁶¹;
- *spInsertBtoA(spA,Iv,IJv,spvF)*⁶²;
- *spSisestaArv(spA,iv,jv,sv)*⁶³;
- *InsertBtoA(A,I,J,IM,JN,B,M,N)*⁶⁴.

Function D.6 (*SsjoudGrbrTalaPunktis(VardaNr,X,AlgPar,lvarras,esFjoud,esQkoormus,suurused)*)^{65 238} – is used to compute the displacements and internal forces of the element 'VardaNr' at $x = X$.

Called function:

- *ESTSTKrmus.m*.

Program D.7 (*spESTGerberBeamWFI.m*)^{66 115} – is used to compute the displacements and internal forces of a Gerber beam.

Called functions:

- *LaheBeamDFI.m*;
- *SisejoudTalaPunktis.m*.

Program D.8 (*spESTbeamLaheWFI.m*)^{67 83} – is used to compute the displacements and internal forces of a beam.

Called functions:

- *LaheBeamDFI.m*;
- *SisejoudTalaPunktis.m*.

Function D.7 (*LaheBeamDFI(baasi0,Ntoerks,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)*)^{68 86 239} – is used to assemble and solve the boundary problem equations of a beam.

Called functions:

- *VardaPikkusT(NSARV,NEARV,krdn,selem)*⁶⁹;
- *yspTlvfmhvI(baasi0,x,l,GAr,EJ)*⁷⁰;
- *yspTlfhlin(baasi0,x,GAr,EJ)*⁷¹;
- *ESTtalaKrmus(baasi0,xx,Li,Fjoud,qkoormus,EI)*⁷²;

⁵⁹ `./octavePrograms/SpTeisendusUhhkMaatriks1x0.m`

⁶⁰ `./octavePrograms/ylSTfhlin.m`

⁶¹ `./octavePrograms/SsjoudGrbrTalaPunktis.m`

⁶² `./octavePrograms/spInsertBtoA.m`

⁶³ `./octavePrograms/spSisestaArv.m`

⁶⁴ `./octavePrograms/InsertBtoA.m`

⁶⁵ `./octavePrograms/SsjoudGrbrTalaPunktis.m`

⁶⁶ `./octavePrograms/spESTGerberBeamWFI.m`

⁶⁷ `./octavePrograms/spESTbeamLaheWFI.m`

⁶⁸ `./octavePrograms/LaheBeamDFI.m`

⁶⁹ `./octavePrograms/VardaPikkusT.m`

⁷⁰ `./octavePrograms/yspTlvfmhvI.m`

⁷¹ `./octavePrograms/yspTlfhlin.m`

⁷² `./octavePrograms/ESTtalaKrmus.m`

- `yzTfzv(baasi0,x,a,Fz,EJ)` ⁷³;
- `yzThqz(baasi0,x,qz,EJ)` ⁷⁴;
- `VardadSolmesT(NSARV,NEARV,Solm,AB,ABB)` ⁷⁵;
- `SpTeisendusUhiKMaatriks1x0(VarrasN)` ⁷⁶;
- `SpTeisendusUhiKMaatriks2x2(VarrasN)` ⁷⁷;
- `SpToeSiirdeWvektorT(VarrasN)` ⁷⁸;
- `SpToeSiirdeFiVektorT(VarrasN)` ⁷⁹;
- `ylTfhlin(baasi0,x,GAr,EJ)` ⁸⁰;
- `SisejoudTalaPunktis(VardaNr,X,AlgPar,lvarras,selem,esFjoud,esQkoormus;suurused)` ⁸¹;
- `spInsertBtoA(spA,Iv,IJv,spvF)` ⁸²;
- `spSisestaArv(spA,iv,jv,sv)` ⁸³;
- `InsertBtoA(A,I,J,IM,JN,B,M,N)` ⁸⁴.

Function D.8 (`SisejoudTalaPunktis(VardaNr,X,AlgPar,lvarras,selem,esFjoud,esQkoormus,suurused)`) ⁸⁵ 239 – is used to compute the displacements and internal forces of the element 'VardaNr' at $x = X$.

Called function:

- *ESTtalaKrmus.m*.

Function D.9 (`ESTtalaKrmus(baasi0,xx,Li,Fjoud,qkoormus,EI)`) ⁸⁶ 240 – is used to compute the loading vector ($q + F$) for a continuous beam.

Called functions:

- `yzThqz(baasi0,x,qz,EJ)` ⁸⁷;
- `yzTfzv(baasi0,x,a,Fz,EJ)` ⁸⁸.

Program D.9 (`spESTtrussLaheWFI.m`) ⁸⁹ 96 – is used to compute the displacements and internal forces of a plane truss.

Called function:

- *LaheTrussDFI.m*.

⁷³ `./octavePrograms/yzTfzv.m`
⁷⁴ `./octavePrograms/yzThqz.m`
⁷⁵ `./octavePrograms/VardadSolmesT.m`
⁷⁶ `./octavePrograms/SpTeisendusUhiKMaatriks1x0.m`
⁷⁷ `./octavePrograms/SpTeisendusUhiKMaatriks2x2.m`
⁷⁸ `./octavePrograms/SpToeSiirdeWvektorT.m`
⁷⁹ `./octavePrograms/SpToeSiirdeFiVektorT.m`
⁸⁰ `./octavePrograms/ylTfhlin.m`
⁸¹ `./octavePrograms/SisejoudTalaPunktis.m`
⁸² `./octavePrograms/spInsertBtoA.m`
⁸³ `./octavePrograms/spSisestaArv.m`
⁸⁴ `./octavePrograms/InsertBtoA.m`
⁸⁵ `./octavePrograms/SisejoudTalaPunktis.m`
⁸⁶ `./octavePrograms/ESTtalaKrmus.m`
⁸⁷ `./octavePrograms/yzThqz.m`
⁸⁸ `./octavePrograms/yzTfzv.m`
⁸⁹ `./octavePrograms/spESTtrussLaheWFI.m`

Function D.10 (`LaheTrussDFI(baasi0,Ntoerkts,sSolmF,tsolm,tSiire,krdn,selem)`)⁹⁰ 98, 240 – is used to assemble and solve the boundary problem equations for a truss.

Called functions:

- `VardaPikkusTr(NSARV,NEARV,krdn,selem)`⁹¹;
- `yspSRmhvI(baasi0,x,EA)`⁹²;
- `yspSRhlin(baasi0,x,EA)`⁹³;
- `VardadSolmesTr(NSARV,NEARV,Solm,AB,ABB)`⁹⁴;
- `SpTeisendusMaatriksTr2x2(NSARV,NEARV,VarrasN,krdn,selem)`⁹⁵;
- `SpTeisendusMaatriksTr2x1(NSARV,NEARV,VarrasN,krdn,selem)`⁹⁶;
- `SpTeisendusUhikMaatriks0x1v(VarrasN)`⁹⁷;
- `SpTeisendusUhikMaatriks2x2(VarrasN)`⁹⁸;
- `SpToeSiirdeUvektorTr(NSARV,NEARV,VarrasN,krdn,selem)`⁹⁹;
- `SpToeSiirdeWvektorTr(NSARV,NEARV,VarrasN,krdn,selem)`¹⁰⁰;
- `spInsertBtoAvect(spA,IM,JN,spB)`¹⁰¹;
- `spInsertBtoA(spA,IJv,IJv,spvF)`¹⁰²;
- `spSisestaArv(spA,iv,jv,sv)`¹⁰³;
- `InsertBtoA(A,I,J,IM,JN,B,M,N)`¹⁰⁴.

Program D.10 (`spESTtrussN2.m`)¹⁰⁵ 117 – is used to compute the internal forces of a plane truss.

Called functions:

- `spSisestaArv(spA,iv,jv,sv)`¹⁰⁶;
- `spInsertBtoA(spA,IJv,IJv,spvF)`¹⁰⁷.

Function D.11 (`spInsertBtoA(spA,IM,JN,spB)`)¹⁰⁸ 32, 43, 47, 216 – inserts a sparse matrix `spB` into a sparse matrix `spA`, starting at row index `IM` and column index `JN`. The overlapping elements of the matrices `spA` and `spB` are added together.

⁹⁰ `./octavePrograms/LaheTrussDFI.m`

⁹¹ `./octavePrograms/VardaPikkusTr.m`

⁹² `./octavePrograms/yspSRmhvI.m`

⁹³ `./octavePrograms/yspSRhlin.m`

⁹⁴ `./octavePrograms/VardadSolmesTr.m`

⁹⁵ `./octavePrograms/SpTeisendusMaatriksTr2x2.m`

⁹⁶ `./octavePrograms/SpTeisendusMaatriksTr2x1.m`

⁹⁷ `./octavePrograms/SpTeisendusUhikMaatriks0x1v.m`

⁹⁸ `./octavePrograms/SpTeisendusUhikMaatriks2x2.m`

⁹⁹ `./octavePrograms/SpToeSiirdeUvektorTr.m`

¹⁰⁰ `./octavePrograms/SpToeSiirdeWvektorTr.m`

¹⁰¹ `./octavePrograms/spInsertBtoAvect.m`

¹⁰² `./octavePrograms/spInsertBtoA.m`

¹⁰³ `./octavePrograms/spSisestaArv.m`

¹⁰⁴ `./octavePrograms/InsertBtoA.m`

¹⁰⁵ `./octavePrograms/spESTtrussN2.m`

¹⁰⁶ `./octavePrograms/spSisestaArv.m`

¹⁰⁷ `./octavePrograms/spInsertBtoA.m`

¹⁰⁸ `./octavePrograms/spInsertBtoA.m`

Function D.12 (`InsertBtoA(A,I,J,IM,JN,B,M,N)`)¹⁰⁹ 33 – inserts matrix B (dimensions M, N) into matrix A (dimensions I, J), starting at row index IM and column index JN .

Function D.13 (`spSisestaArv(spA,iv,jv,sv)`)¹¹⁰ 45, 218 – inserts the number sv into sparse matrix spA , starting at row index iv and column index jv .

Function D.14 (`SpToeSiirdeUvektor(NSARV,NEARV,VarrasS,krdn,selem)`)¹¹¹ 47 – transforms a displacement of an element in local coordinates into a displacement Ux at the support in global coordinates.

Function D.15 (`SpToeSiirdeWvektor(NSARV,NEARV,VarrasS,krdn,selem)`)¹¹² 47 – transforms a displacement of an element in local coordinates into a displacement Wx at the support in global coordinates.

Function D.16 (`SpToeSiirdeFiVektor(VarrasN)`)¹¹³ 47 – transforms a displacement of an element in local coordinates into a displacement Fi at the support in global coordinates.

Function D.17 (`ylfhlin(baasi0,x,EA,GAr,EJ)`)¹¹⁴ 227 – is used to compute the transfer matrix for a frame (Sign Convention 2). Here,
 EA – axial stiffness of the element,
 GAr – shear stiffness of the element,
 EJ – bending stiffness of the element,
 $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

Function D.18 (`yzhqz(baasi0,x,qx,qz,EA,EJ)`)¹¹⁵ 27, 33, 228 – is used to compute the loading vector of a uniformly distributed load for a frame. Here,
 qx – uniformly distributed load along the x -axis,
 qz – uniformly distributed load along the z -axis,
 EA – axial stiffness of the element,
 EJ – bending stiffness of the element,
 $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

Function D.19 (`yzThqz(baasi0,x,qz,EJ)`)¹¹⁶ 228 – is used to compute the loading vector of a uniformly distributed load for a beam. Here,
 qz – uniformly distributed load along the z -axis,
 EJ – bending stiffness of the element,
 $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

¹⁰⁹ `./octavePrograms/InsertBtoA.m`

¹¹⁰ `./octavePrograms/spSisestaArv.m`

¹¹¹ `./octavePrograms/SpToeSiirdeUvektor.m`

¹¹² `./octavePrograms/SpToeSiirdeWvektor.m`

¹¹³ `./octavePrograms/SpToeSiirdeFiVektor.m`

¹¹⁴ `./octavePrograms/ylfhlin.m`

¹¹⁵ `./octavePrograms/yzhqz.m`

¹¹⁶ `./octavePrograms/yzThqz.m`

Function D.20 (`yzShqz(x,qx,qz)`)¹¹⁷ 228 – is used to compute the loading vector of a uniformly distributed load for a three-hinged frame. Here,
 qx – uniformly distributed load along the x -axis,
 qz – uniformly distributed load along the z -axis.

Function D.21 (`yzSThqz(x,qz)`)¹¹⁸ 228 – is used to compute the loading vector of a uniformly distributed load for a Gerber beam. Here,
 qz – uniformly distributed load along the z -axis.

Function D.22 (`yzfzv(baasi0,x,a,Fx,Fz,EA,EJ)`)¹¹⁹ 27, 33, 228 – is used to compute the loading vector of the point load for a frame. Here,
 Fx – point load acting in the x direction,
 Fz – point load acting in the z direction,
 a – distance of a load point from a start point,
 EA – axial stiffness of the element,
 EJ – bending stiffness of the element,
 $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

Function D.23 (`yzmyv(baasi0,x,a,My,EJ)`)¹²⁰ 27 – is used to compute the loading vector of a moment load. Here,
 My – load along the y -axis.

Function D.24 (`yzTfzv(baasi0,x,a,Fz,EJ)`)¹²¹ 228 – is used to compute the loading vector of a point load for a beam. Here,
 Fz – point load acting in the z direction,
 a – distance of a load point from a start point,
 EJ – bending stiffness of the element,
 $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

Function D.25 (`yzSfzv(x,a,Fx,Fz)`)¹²² 228 – is used to compute the loading vector of a point load for a three-hinged frame. Here,
 Fx – point load acting in the x direction,
 Fz – point load acting in the z direction,
 a – distance of a load point from a start point.

Function D.26 (`yzSTfzv(x,a,Fz)`)¹²³ 228 – is used to compute the loading vector of a point load for a Gerber beam. Here,
 Fz – point load acting in the z direction,
 a – distance of a load point from a start point.

¹¹⁷ [./octavePrograms/yzShqz.m](#)

¹¹⁸ [./octavePrograms/yzSThqz.m](#)

¹¹⁹ [./octavePrograms/yzfzv.m](#)

¹²⁰ [./octavePrograms/yzmyv.m](#)

¹²¹ [./octavePrograms/yzTfzv.m](#)

¹²² [./octavePrograms/yzSfzv.m](#)

¹²³ [./octavePrograms/yzSTfzv.m](#)

Function D.27 (`ysplfhlin(baasi0,x,EA,GAr,EJ)`)¹²⁴ 227 – is used to compute the transfer matrix for a frame (Sign Convention 2). Here,

EA – axial stiffness of the element,

GA_r – shear stiffness of the element,

EJ – bending stiffness of the element,

baasi0 – scaling multiplier for the displacements ($i = EJ/l$).

Function D.28 (`yspTlfhlin(baasi0,x,GAr,EJ)`)¹²⁵ 227 – is used to compute the transfer matrix for a beam (Sign Convention 2). Here,

GA_r – shear stiffness of the element,

EJ – bending stiffness of the element,

baasi0 – scaling multiplier for the displacements ($i = EJ/l$).

Function D.29 (`yspSlfhlin(x)`)¹²⁶ 227 – is used to compute the transfer matrix for a three-hinged frame (Sign Convention 2).

Function D.30 (`yspSTlfhlin(x)`)¹²⁷ 227 – is used to compute the transfer matrix for a Gerber beam (Sign Convention 2).

Function D.31 (`ysplvfmhvI(baasi0,x,l,EA,GAr,EJ)`)¹²⁸ 33 – is used to compute the sparse transfer matrix $\widehat{\mathbf{U}}$ for a frame (Sign Convention 2). Here,

EA – axial stiffness of the element,

GA_r – shear stiffness of the element,

EJ – bending stiffness of the element,

baasi0 – scaling multiplier for the displacements ($i = EJ/l$).

Called functions:

- `ysplfhlin(baasi0,x,EA,GAr,EJ)` ¹²⁹;
- `spInsertBtoA(spA,IM,JN,spB)` ¹³⁰.

Function D.32 (`yspTlvfmhvI(baasi0,x,l,GAr,EJ)`)¹³¹ 227 – is used to compute the sparse transfer matrix for a beam (Sign Convention 2). Here,

GA_r – shear stiffness of the element,

EJ – bending stiffness of the element,

baasi0 – scaling multiplier for the displacements ($i = EJ/l$).

Called functions:

- `yspTlfhlin(baasi0,x,GAr,EJ)` ¹³²;
- `spInsertBtoA(spA,IM,JN,spB)` ¹³³.

¹²⁴ `./octavePrograms/ysplfhlin.m`

¹²⁵ `./octavePrograms/yspTlfhlin.m`

¹²⁶ `./octavePrograms/yspSlfhlin.m`

¹²⁷ `./octavePrograms/yspSTlfhlin.m`

¹²⁸ `./octavePrograms/ysplvfmhvI.m`

¹²⁹ `./octavePrograms/ysplfhlin.m`

¹³⁰ `./octavePrograms/spInsertBtoA.m`

¹³¹ `./octavePrograms/yspTlvfmhvI.m`

¹³² `./octavePrograms/yspTlfhlin.m`

¹³³ `./octavePrograms/spInsertBtoA.m`

Function D.33 (`yspSlvfmhvI(x)`)^{134 227} – is used to compute the transfer matrix for a three-hinged frame (Sign Convention 2).

Called functions:

- `yspSlfhlin(x)` ¹³⁵;
- `spInsertBtoA(spA,IM,JN,spB)` ¹³⁶.

Function D.34 (`yspSTlvfmhvI(x)`)^{137 227} – is used to compute the sparse transfer matrix for a Gerber beam (Sign Convention 2).

Called functions:

- `yspSTlfhlin(x)` ¹³⁸;
- `spInsertBtoA(spA,IM,JN,spB)` ¹³⁹.

Function D.35 (`yspSRmhvI(baasi0,Li,EA)`)^{140 53} – is used to compute the sparse transfer matrix for a truss.

Called functions:

- `yspSRhlin(baasi0,x,EA)` ¹⁴¹;
- `spInsertBtoA(spA,IM,JN,spB)` ¹⁴².

Function D.36 (`yspSRhlin(baasi0,x,EA)`)^{143 53} – is used to compute the sparse transfer matrix for a truss. Here, EA – axial stiffness of the element, $baasi0$ – scaling multiplier for the displacements ($i = EJ/l$).

Function D.37 (`ESTtalaKrmus(baasi0,xx,Li,Fjoud,qkoormus,EI)`)^{144 228} – is used to compute the loading vector ($q + F$) for a continuous beam.

Called functions:

- `yzThqz(baasi0,x,qz,EJ)` ¹⁴⁵;
- `yzTfzv(baasi0,x,a,Fz,EJ)` ¹⁴⁶.

Function D.38 (`ESTSKrmus(xx,Li,Fjoud,qkoormus)`)^{147 228} – is used to compute the loading vector ($q + F$) for a three-hinged frame.

Called functions:

- `yzShqz(x,qx,qz)` ¹⁴⁸;
- `yzSfzv(x,a,Fx,Fz)` ¹⁴⁹.

¹³⁴ `./octavePrograms/yspSlvfmhvI.m`
¹³⁵ `./octavePrograms/yspSlfhlin.m`
¹³⁶ `./octavePrograms/spInsertBtoA.m`
¹³⁷ `./octavePrograms/yspSTlvfmhvI.m`
¹³⁸ `./octavePrograms/yspSlfhlin.m`
¹³⁹ `./octavePrograms/spInsertBtoA.m`
¹⁴⁰ `./octavePrograms/yspSRmhvI.m`
¹⁴¹ `./octavePrograms/yspSRhlin.m`
¹⁴² `./octavePrograms/spInsertBtoA.m`
¹⁴³ `./octavePrograms/yspSRhlin.m`
¹⁴⁴ `./octavePrograms/ESTtalaKrmus.m`
¹⁴⁵ `./octavePrograms/yzThqz.m`
¹⁴⁶ `./octavePrograms/yzTfzv.m`
¹⁴⁷ `./octavePrograms/ESTSKrmus.m`
¹⁴⁸ `./octavePrograms/yzShqz.m`
¹⁴⁹ `./octavePrograms/yzSfzv.m`

Function D.39 (`ESTSTKrmus(xx, Li, Fjoud, qkoormus)`)¹⁵⁰ 239 – is used to compute the loading vector ($q + F$) for a Gerber beam.

Called functions:

- `yzSThqz(x, qz)` ¹⁵¹;
- `yzSTfzv(x, a, Fz)` ¹⁵².

Program D.11 (`spESTframe50WFI.m`)¹⁵³ – is used to compute the displacements and internal forces of a plane frame.

Called function:

- `LaheFrameDFIm.m` ¹⁵⁴.

Program D.12 (`spESTframe3hinge1WFI.m`)¹⁵⁵ – is used to compute the displacements and internal forces of a plane frame.

Called function:

- `LaheFrameDFIm.m` ¹⁵⁶.

Program D.13 (`spESTframe3hinge1NQM.m`)¹⁵⁷ – is used to compute the internal forces of a plane frame.

Called function:

- `LaheFrame3hingeNQM.m` ¹⁵⁸.

D.2 Programs for second-order analysis

Program D.14 (`yspESTframe2LaheWFI.m`)¹⁵⁹ 158 – is used to compute the displacements and forces of a plane frame.

Called function:

- `Lahe2FrameDFIm.m`.

Function D.40 (`Lahe2FrameDFIm(baasi0, Ntoerks, esQkoormus, esFjoud, sSolmF, tsolm, tSiire, krnd, selem)`)¹⁶⁰ 148, 152, 160, 246, 248 – is used to assemble and solve the boundary problem equations for a plane frame.

Called functions:

- `VardaPikkus(NSARV, NEARV, krnd, selem)` ¹⁶¹;
- `ysplvmhvII(baasi0, S, x, l, EA, GAr, EJ)` ¹⁶²;

¹⁵⁰ `./octavePrograms/ESTSTKrmus.m`

¹⁵¹ `./octavePrograms/yzSThqz.m`

¹⁵² `./octavePrograms/yzSTfzv.m`

¹⁵³ `./octavePrograms/spESTframe50WFI.m`

¹⁵⁴ `./octavePrograms/LaheFrameDFIm.m`

¹⁵⁵ `./octavePrograms/spESTframe3hinge1WFI.m`

¹⁵⁶ `./octavePrograms/LaheFrameDFIm.m`

¹⁵⁷ `./octavePrograms/spESTframe3hinge1NQM.m`

¹⁵⁸ `./octavePrograms/LaheFrame3hingeNQM.m`

¹⁵⁹ `./octavePrograms/yspESTframe2LaheWFI.m`

¹⁶⁰ `./octavePrograms/Lahe2FrameDFIm.m`

¹⁶¹ `./octavePrograms/VardaPikkus.m`

¹⁶² `./octavePrograms/ysplvmhvII.m`

- *ysplfmhvII*(*baasi0*,*S*,*x*,*l*,*EA*,*GAr*,*EJ*) ¹⁶³;
- *tnnusarv*(*l1*,*s1*,*ea1*) ¹⁶⁴;
- *ysplfhlin*(*baasi0*,*x*,*EA*,*GAr*,*EJ*) ¹⁶⁵;
- *ylqvII*(*baasi0*,*S*,*x*,*lp*,*qz*,*EA*,*EJ*) ¹⁶⁶;
- *yzhqz*(*baasi0*,*x*,*qx*,*qz*,*EA*,*EJ*) ¹⁶⁷;
- *ylfhvzII*(*baasi0*,*S*,*x*,*l*,*a*,*Fz*,*EJ*) ¹⁶⁸;
- *yzfzv*(*baasi0*,*x*,*a*,*Fx*,*Fz*,*EA*,*EJ*) ¹⁶⁹;
- *VardadSolmes*(*NSARV*,*NEARV*,*Solm*,*AB*,*ABB*) ¹⁷⁰;
- *SpTeisendusMaatriks*(*NSARV*,*NEARV*,*VarrasN*,*krdn*,*selem*) ¹⁷¹;
- *SpTeisendusMaatriks2x2*(*NSARV*,*NEARV*,*VarrasN*,*krdn*,*selem*) ¹⁷²;
- *SpTeisendusUhikMaatriks2x2*(*VarrasN*) ¹⁷³;
- *SpTeisendusUhikMaatriks*(*VarrasN*) ¹⁷⁴;
- *SpTeisendusUhikMaatriks0x1v*(*VarrasN*) ¹⁷⁵;
- *SpToeSiirdeUvektor*(*NSARV*,*NEARV*,*VarrasN*,*krdn*,*selem*) ¹⁷⁶;
- *SpToeSiirdeWvektor*(*NSARV*,*NEARV*,*VarrasN*,*krdn*,*selem*) ¹⁷⁷;
- *SpToeSiirdeFiVektor*(*VarrasN*) ¹⁷⁸;
- *ylfmhvII*(*baasi0*,*S*,*x*,*l*,*EA*,*GAr*,*EJ*) ¹⁷⁹;
- *ytransfp*(*S*,*l*,*EJ*) ¹⁸⁰;
- *spSisestaArv*(*spA*,*iv*,*ju*,*sv*) ¹⁸¹;
- *spInsertBtoA*(*spA*,*Iv*,*IJv*,*spvF*) ¹⁸²;
- *InsertBtoA*(*A*,*I*,*J*,*IM*,*JN*,*B*,*M*,*N*) ¹⁸³.

163. `./octavePrograms/ysplfmhvII.m`
164. `./octavePrograms/tnnusarv.m`
165. `./octavePrograms/ysplfhlin.m`
166. `./octavePrograms/ylqvII.m`
167. `./octavePrograms/yzhqz.m`
168. `./octavePrograms/ylfhvzII.m`
169. `./octavePrograms/yzfzv.m`
170. `./octavePrograms/VardadSolmes.m`
171. `./octavePrograms/SpTeisendusMaatriks.m`
172. `./octavePrograms/SpTeisendusMaatriks2x2.m`
173. `./octavePrograms/SpTeisendusUhikMaatriks2x2.m`
174. `./octavePrograms/SpTeisendusUhikMaatriks.m`
175. `./octavePrograms/SpTeisendusUhikMaatriks0x1v.m`
176. `./octavePrograms/SpToeSiirdeUvektor.m`
177. `./octavePrograms/SpToeSiirdeWvektor.m`
178. `./octavePrograms/SpToeSiirdeFiVektor.m`
179. `./octavePrograms/ylfmhvII.m`
180. `./octavePrograms/ytransfp.m`
181. `./octavePrograms/spSisestaArv.m`
182. `./octavePrograms/spInsertBtoA.m`
183. `./octavePrograms/InsertBtoA.m`

Program D.15 (`yspESTframe1LaheWFI.m`)¹⁸⁴ 150 – is used to compute the displacements and internal forces for a plane frame.

Called function:

- `Lahe2FrameDFIm.m`.

Function D.41 (`ylfmhvII(baasi0,S,x,l,EA,GAr,EJ)`)¹⁸⁵ 144, 228 – is used to compute the transfer matrix for a frame (Sign Convention 2) in a second-order analysis. Here,

EA – axial stiffness of the element,

GAr – shear stiffness of the element,

EJ – bending stiffness of the element,

l – length of the element,

$baasi0$ – scaling multiplier for the displacements ($i = EJ/l$),

S – axial force.

Called functions:

- `tnnusrv(l1,s1,ea1)` ¹⁸⁶;
- `ylfhlin(baasi0,x,EA,GAr,EJ)` ¹⁸⁷.

Function D.42 (`ysplfmhvII(baasi0,S,x,l,EA,GAr,EJ)`)¹⁸⁸ 233 – is used to compute the transfer matrix for a frame (Sign Convention 2) in a second-order analysis. Here,

is used to assemble and solve the boundary problem equations for a plane frame. EA – axial stiffness of the element,

GAr – shear stiffness of the element,

EJ – bending stiffness of the element,

l – length of the element,

$baasi0$ – scaling multiplier for the displacements ($i = EJ/l$),

S – axial force.

Called functions:

- `tnnusrv(l1,s1,ea1)` ¹⁸⁹;
- `ysplfhlin(baasi0,x,EA,GAr,EJ)` ¹⁹⁰.

Function D.43 (`ylfhvzII(baasi0,S,l,x,a,Fz,EJ)`)¹⁹¹ 145, 233 – is used to compute the loading vector of a point load for a frame in a second-order analysis. Here,

Fz – point load acting in the z direction,

a – distance of a load point from a start point,

EJ – bending stiffness of the element,

l – length of the element,

$baasi0$ – scaling multiplier for the displacements ($i = EJ/l$),

S – axial force.

Called functions:

¹⁸⁴ `./octavePrograms/yspESTframe1LaheWFI.m`

¹⁸⁵ `./octavePrograms/ylfmhvII.m`

¹⁸⁶ `./octavePrograms/tnnusrv.m`

¹⁸⁷ `./octavePrograms/ylfhlin.m`

¹⁸⁸ `./octavePrograms/ysplfmhvII.m`

¹⁸⁹ `./octavePrograms/tnnusrv.m`

¹⁹⁰ `./octavePrograms/ysplfhlin.m`

¹⁹¹ `./octavePrograms/ylfhvzII.m`

- `tnnusrv(l1,s1,ea1)` ¹⁹²;
- `yzfzv(baasi0,x,a,Fx,Fz,EA,EJ)` ¹⁹³.

Function D.44 (`ylqvII(baasi0,S,x,lp,qz,EJ)`) ¹⁹⁴ 233 – is used to compute the loading vector of a uniformly distributed load for a frame in a second-order analysis. Here, `qz` – uniformly distributed load along the z-axis, `EJ` – bending stiffness of the element, `baasi0` – scaling multiplier for the displacements ($i = EJ/l$), `S` – axial force.

Called functions:

- `tnnusrv(l1,s1,ea1)` ¹⁹⁵;
- `yzhqz(baasi0,x,qx,qz,EA,EJ)` ¹⁹⁶.

Function D.45 (`ylfmII(S,x,l,EA,GAr,EJ)`) ¹⁹⁷ 140, 234 – is used to compute the transfer matrix for a frame (Sign Convention 2) in a second-order analysis. Here, `EA` – axial stiffness of the element, `GAr` – shear stiffness of the element, `EJ` – bending stiffness of the element, `S` – axial force.

Called functions:

- `tnnusrv(l1,s1,ea1)` ¹⁹⁸;
- `ylflin(x,EA,GAr,EJ)` ¹⁹⁹.

Function D.46 (`ylqII(S,x,l,qz,EJ)`) ²⁰⁰ 141 – is used to compute the loading vector of a uniformly distributed load in a second-order analysis. Here, `qz` – uniformly distributed load along the x-axis, `EJ` – bending stiffness of the element, `l` – length of the element, `S` – axial force.

Called functions:

- `tnnusrv(l1,s1,ea1)` ²⁰¹;
- `yzqz(x,q,EJ)` ²⁰².

Function D.47 (`ylffzII(S,l,x,a,Fz,EJ)`) ²⁰³ 141 – is used to compute the loading vector for the point load in a second-order analysis. Here, `Fz` – point load acting in the z direction,

¹⁹² `./octavePrograms/tnnusrv.m`

¹⁹³ `./octavePrograms/yzfzv.m`

¹⁹⁴ `./octavePrograms/ylqvII.m`

¹⁹⁵ `./octavePrograms/tnnusrv.m`

¹⁹⁶ `./octavePrograms/yzhqz.m`

¹⁹⁷ `./octavePrograms/ylfmII.m`

¹⁹⁸ `./octavePrograms/tnnusrv.m`

¹⁹⁹ `./octavePrograms/ylflin.m`

²⁰⁰ `./octavePrograms/ylqII.m`

²⁰¹ `./octavePrograms/tnnusrv.m`

²⁰² `./octavePrograms/yzqz.m`

²⁰³ `./octavePrograms/ylffzII.m`

a – distance of a load point from a start point,

EJ – bending stiffness of the element,

l – length of the element,

S – axial force.

Called functions:

– `tnnusrv(l1,s1,ea1)` ²⁰⁴;

– `yzfz(x,a,Fx,Fz,EA,EJ)` ²⁰⁵.

Function D.48 (ytransf(S,l,EJ)) ²⁰⁶ 143 – is used to compute a transformation matrix to transform the forces N and Q into the forces S and H in a second-order analysis. Here,

EJ – bending stiffness of the element,

l – length of the element,

S – axial force.

Called function:

– `tnnusrv(l1,s1,ea1)` ²⁰⁷.

Function D.49 (ytransfp(S,l,EJ)) ²⁰⁸ 143 – is used to compute a transformation matrix to transform the forces S and H into the forces N and Q in a second-order analysis. Here,

EJ – bending stiffness of the element,

l – length of the element,

S – axial force.

Called function:

– `tnnusrv(l1,s1,ea1)` ²⁰⁹.

Function D.50 (ysplvmhvII(baasi0,S,x,l,EA,GAr,EJ)) ²¹⁰ 233 – is used to compute the sparse transfer matrix $\widehat{\mathbf{IU}}$ ($I \cdot Z_p - U \cdot Z_v = Z_0$, $Z_p' = [U_p \ W_p \ Fip \ Sp \ Hp \ Mp]$) for a frame (Sign Convention 2) in a second-order analysis. Here,

EA – axial stiffness of the element,

GAr – shear stiffness of the element,

EJ – bending stiffness of the element,

l – length of the element,

$baasi0$ – scaling multiplier for the displacements ($i = EJ/l$),

S – S – axial force.

Called functions:

– `ysplvmhvII(baasi0,S,x,l,EA,GAr,EJ)` ²¹¹;

– `spInsertBtoA(spA,IM,JN,spB)` ²¹².

²⁰⁴ `./octavePrograms/tnnusrv.m`

²⁰⁵ `./octavePrograms/yzfz.m`

²⁰⁶ `./octavePrograms/ytransf.m`

²⁰⁷ `./octavePrograms/tnnusrv.m`

²⁰⁸ `./octavePrograms/ytransfp.m`

²⁰⁹ `./octavePrograms/tnnusrv.m`

²¹⁰ `./octavePrograms/ysplvmhvII.m`

²¹¹ `./octavePrograms/ysplvmhvII.m`

²¹² `./octavePrograms/spInsertBtoA.m`

D.3 Programs for plastic analysis

Program D.16 (`spESTframeSn3LaheWFI.m`)²¹³ 182 – is used to compute the displacements and internal forces of a plane frame.

Called function:

- `LaheFrameSnDFIm.m`.

Function D.51 (`LaheFrameSnDFIm(baasi0,Ntoerkts,esQkoormus,esFjoud,sSolmF,tsolm,tSiire,krdn,selem)`)²¹⁴ 182, 188, 186, 195, 199, 201, 205, 251, 252

– is used to assemble and solve the boundary problem equations of a plane frame.

Called functions:

- `ESTFrKrmus(baasi0,xx,Li,Fjoud,qkoormus,EA,EI)`²¹⁵;
- `ysplvmhvI(baasi0,x,l,EA,GAr,EJ)`²¹⁶;
- `ysplfhlin(baasi0,x,EA,GAr,EJ)`²¹⁷;
- `yzhqzm(baasi0,x,a,qx,qz,EA,EJ)`²¹⁸;
- `yzfzv(baasi0,x,a,Fx,Fz,EA,EJ)`²¹⁹;
- `VardadSolmes(NSARV,NEARV,Solm,AB,ABB)`²²⁰;
- `VardaPikkus(NSARV,NEARV,krdn,selem)`²²¹;
- `SpTeisendusMaatriks(NSARV,NEARV,VarrasN,krdn,selem)`²²²;
- `SpTeisendusMaatriks2x2(NSARV,NEARV,VarrasN,krdn,selem)`²²³;
- `SpTeisendusUhiKMaatriks2x2(VarrasN)`²²⁴;
- `SpTeisendusUhiKMaatriks(VarrasN)`²²⁵;
- `SpTeisendusUhiKMaatriks0x1v(VarrasN)`²²⁶;
- `SpToeSiirdeUvektor(NSARV,NEARV,VarrasN,krdn,selem)`²²⁷;
- `SpToeSiirdeWvektor(NSARV,NEARV,VarrasN,krdn,selem)`²²⁸;
- `SpToeSiirdeFiVektor(VarrasN)`²²⁹;
- `ylfhlin(baasi0,x,EA,GAr,EJ)`²³⁰;
- `spSisestaArv(spA,iv,jv,sv)`²³¹;

²¹³ `./octavePrograms/spESTframeSn3LaheWFI.m`

²¹⁴ `./octavePrograms/LaheFrameSnDFIm.m`

²¹⁵ `./octavePrograms/ESTFrKrmus.m`

²¹⁶ `./octavePrograms/ysplvmhvI.m`

²¹⁷ `./octavePrograms/ysplfhlin.m`

²¹⁸ `./octavePrograms/yzhqzm.m`

²¹⁹ `./octavePrograms/yzfzv.m`

²²⁰ `./octavePrograms/VardadSolmes.m`

²²¹ `./octavePrograms/VardaPikkus.m`

²²² `./octavePrograms/SpTeisendusMaatriks.m`

²²³ `./octavePrograms/SpTeisendusMaatriks2x2.m`

²²⁴ `./octavePrograms/SpTeisendusUhiKMaatriks2x2.m`

²²⁵ `./octavePrograms/SpTeisendusUhiKMaatriks.m`

²²⁶ `./octavePrograms/SpTeisendusUhiKMaatriks0x1v.m`

²²⁷ `./octavePrograms/SpToeSiirdeUvektor.m`

²²⁸ `./octavePrograms/SpToeSiirdeWvektor.m`

²²⁹ `./octavePrograms/SpToeSiirdeFiVektor.m`

²³⁰ `./octavePrograms/ylfhlin.m`

²³¹ `./octavePrograms/spSisestaArv.m`

- *spInsertBtoA*(*spA*,*I**v*,*I**J**v*,*spvF*)²³²;
- *InsertBtoA*(*A*,*I*,*J*,*IM*,*JN*,*B*,*M*,*N*)²³³.

Program D.17 (*spESTframeSn2LaheWFI.m*)²³⁴ 189 – is used to compute the displacements and internal forces of a plane frame.

Called function:

- *LaheFrameSnDFIm.m*.

Program D.18 (*spESTframeSn1LaheWFI.m*)²³⁵ 195 – is used to compute the displacements and internal forces of a plane frame.

Called function:

- *LaheFrameSnDFIm.m*.

Program D.19 (*spESTframeSn0LaheWFI.m*)²³⁶ 202 – is used to compute the displacements and internal forces of a plane frame.

Called function:

- *LaheFrameSnDFIm.m*.

²³² [./octavePrograms/spInsertBtoA.m](#)

²³³ [./octavePrograms/InsertBtoA.m](#)

²³⁴ [./octavePrograms/spESTframeSn2LaheWFI.m](#)

²³⁵ [./octavePrograms/spESTframeSn1LaheWFI.m](#)

²³⁶ [./octavePrograms/spESTframeSn0LaheWFI.m](#)

Bibliography

- [BP13] R. M. Barker, J. A. Puckett. *Design of Highway Bridges: An LRFD Approach*, 3. edition.¹ Hoboken, New Jersey: John Wiley & Sons, 2013. Web. 08 August 2013. [221](#)
- [Bor79a] F. W. Bornscheuer. *Berechnungsverfahren nach Theorie II. Ordnung*. Stuttgart: Institut für Baustatik der Universität Stuttgart, 1979. [225](#)
- [Bor79b] F. W. Bornscheuer. *Übertragungsverfahren*. Stuttgart: Institut für Baustatik der Universität Stuttgart, 1979. [137](#), [227](#), [232](#)
- [BW80] C. A. Brebia, S. Walker. *Boundary Element Techniques in Engineering*. London: Butterworths, 1980. [4](#)
- [Cpr11] C. Caprani. *Plastic Analysis. Structural Engineering*.² Structural Analysis III, 2010/11. Web. 06 October 2013. [169](#), [173](#), [174](#)
- [Din11] D. Dinkler. *Grundlagen der Baustatik. Modelle und Berechnungsmethoden für ebene Stabtragwerke*. 1. Auflage 2011.³ Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2011. Web. 08 August 2013.
- [EBH08] J. W. Eaton, D. Bateman, S. Hauberg. *GNU Octave Manual Version 3. A high-level interactive language for numerical computations. Edition 3 for Octave version 3.0.2*.⁴ Network Theory Ltd, August, 2008. Web. 08 August 2013. [33](#)
- [GN12] M. K. Gíslason, D. H. Nash. Finite Element Modelling of a Multi-Bone Joint: The Human Wrist.⁵ In: Dr. Farzad Ebrahimi (ed.), *Finite Element Analysis – New Trends and Developments*. InTech, 2012. DOI: 10.5772/50560. Web. 08 August 2013. [30](#), [222](#)

¹9.4.1 Betti's Theorem – http://books.google.ee/books?id=ytKw4IZD_ZMC&pg=PA137&dq=betti%27s+theorem+forces+of+first+load+state+1+to+state+2&hl=en&sa=X&ei=j3wDUsSKAsrQsgaUp4GIDA&ved=0CDgQ6AEwAg#v=onepage&q=betti%27s%20theorem%20forces%20of%20first%20load%20state%20%20to%20state%202&f=false

²<http://www.colincaprani.com/files/notes/SAIII/PlasticAnalysis1011.pdf>

³<http://books.google.ee/books?id=0q0aKoH2Y40C&pg=PA33&lpg=PA33&dq=Baustatik+Schnittprinzip&source=bl&ots=n0xKBXI7m&sig=fc8Y6Z3K0VKyEay9wLIP9Bh91S0&hl=et#v=onepage&q=Baustatik%20Schnittprinzip&f=false>

⁴<http://www.network-theory.co.uk/docs/octave3/index.html>

⁵<http://www.intechopen.com/books/finite-element-analysis-new-trends-and-...developments/finite-element-modelling-of-a-multi-bone-joint-the-human-wrist>

- [Har87] F. Hartmann. *Methode der Randelemente*. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1987. 4
- [IB01] M. Jirásek, Z. P. Bažant. Chapter 13. Combined plastic bending and compression or tension.⁶ In: *Inelastic Analysis of Structures*.⁷ John Wiley & Sons, 2001. Web. 06 October 2013. 170
- [KL09] I. A. Karnovsky, O. Lebed. *Advanced Methods of Structural Analysis. Initial Parameters Method*.⁸ Heidelberg, New York: Springer-Verlag, 2009. Web. 21. Jun. 2014. 3
- [KB95] B. Kolman, R. E. Beck. *Elementary Linear Programming with Applications*. 2nd edition.⁹ Academic Press, 1995. Web. 16 August 2013. 35
- [Krä90] W. B. Krätzig. *Statik der Tragwerke, 2b. Das Weggrößenverfahren*. Bochum: Ruhr-Universität Bochum, Fakultät für Bauingenieurwesen, 1990. 4
- [Krä91a] W. B. Krätzig. *Statik der Tragwerke, 4. Finite Berechnungsmethoden (4.a Grundlagen; Stabtragwerke)*. Bochum: Ruhr-Universität Bochum, Fakultät für Bauingenieurwesen, 1991. 4, 137, 225
- [Krä91b] W. B. Krätzig. *Statik der Tragwerke, 4. Computermethoden zur linearen Tragwerksanalyse*. Teil II. Bochum: Ruhr-Universität Bochum, Fakultät für Bauingenieurwesen, 1991. 4, 137, 140, 142, 143, 227, 232
- [Krä91c] W. B. Krätzig. *Tragwerke 2. Theorie und Berechnungsmethoden statisch unbestimmter Stabtragwerke*. Berlin, Heidelberg, New York: Springer-Verlag, 1991. 4
- [KW90] W. B. Krätzig und Wittek. *Tragwerke 1. Theorie und Berechnungsmethoden statisch bestimmter Stabtragwerke*. Berlin, Heidelberg, New York: Springer-Verlag, 1990. 4, 30, 52, 120
- [KHMW10] W. B. Krätzig, R. Harte, K. Meskouris und U. Wittek. *Tragwerke 1. Theorie und Berechnungsmethoden statisch bestimmter Stabtragwerke*. 5., überarbeitete Auflage.¹⁰ Springer-Verlag, 2010. Web. 08 August 2013. 18, 221

⁶<http://www.wiley.com/legacy/wileychi/inelastic/pdf/C13.PDF>

⁷<http://www.wiley.com/legacy/wileychi/inelastic/>

⁸http://books.google.ee/books?id=W2JVDvs_KuEC&pg=PA147&lpg=PA147&dq=initial+parameter+structural+analyses&source=bl&ots=OXt5ctbAZH&sig=Fq0TKZAcLG4S193zLDUhyfu6CBY&hl=en&sa=X&ei=lyClU4aMD40wOQW9qoDADQ&ved=0CDUQ6AEwAw\#v=onepage&q=initial%20parameter%20structural%20analyses&f=false

⁹<http://store.elsevier.com/product.jsp?isbn=9780124179103>

¹⁰2.4.1 Zustandsgrößen – <http://books.google.ee/books?id=itFS-x8fv7oC&pg=PA53&dq=Zustandsgr%C3%B6ssen+Kr%C3%A4tzig&hl=en&sa=X&ei=LSsDUusjfAZH2sgatiIHgCQ&ved=0CCsQ6AEwAA#v=onepage&q=Zustandsgr%C3%B6ssen%20Kr%C3%A4tzig&f=false>

2.4.4 Formänderungsarbeits-Funktionale – <http://books.google.ee/books?id=itFS-x8fv7oC&pg=PA58&dq=Form%C3%A4nderungsarbeits-Funktionale+Kr%C3%A4tzig&hl=en&sa=X\string&ei=wVsDUoWDGsrDswa204HAAw&ved=0CCsQ6AEwAA#v=onepage&q=Form%C3%A4nderungsarbeits-Funktionale%20Kr%C3%A4tzig&f=false>

- [KHMW05] W. B. Krätzig, R. Harte, K. Meskouris und U. Wittek. *Tragwerke 2. Theorie und Berechnungsmethoden statisch unbestimmter Stabtragwerke*. 4., überarbeitete und erweiterte Auflage.¹¹ Springer-Verlag, 2004, 1994, 1998, 2005. Web. 08 August 2013. 18
- [Lah97a] A. Lahe. The transfer matrix and the boundary element method.¹² *Proc. Estonian Acad. Sci. Engng.*, 1997, 3, 1, pp. 3–12. Web. 08 August 2013. 4
- [Lah97b] A. Lahe. The EST method for the frame analysis. In: *Proc. Tenth Nordic Seminar on Computational Mechanics*, October 24–25, 1997. Tallinn: Tallinn Technical University, 1997, pp. 202–205. 4
- [Lah98a] A. Lahe. The EST method for the frame analysis in second order theory, In: *Proc. of the NSCM-11: Nordic Seminar on Computational Mechanics*, October 16–17, 1997. Stockholm: Royal Institute of Technology, Department of Structural Engineering, TRITA-BKN. Bulletin 39, 1998, pp.167–170. 4
- [Lah12] A. Lahe. *Ehitusmehaanika*.¹³ Tallinn: Tallinna Tehnikaülikooli Kirjastus, 2012. 764 lk. Web. 08 August 2013. 62, 77, 94, 120, 166
- [LT80] M. Lawo, O. Tierauf. *Matrizenmethoden der Statik und Dynamik, I: Statik*. Braunschweig-Wiesbaden: Fried. Vieweg & Sohn, 1980. 4
- [MPHS] *Shear and Moment Diagrams for Frames*.¹⁴ The University of Memphis.¹⁵ 16 Web. 25 August 2013. 29, 50, 132
- [OUD10] V. Oudalov. *Minimal spanning and maximal independent sets, Basis and Dimension*.¹⁷ Handout 1, February 23, 2010. Web. 17 August 2013. 35
- [PL63] E. C. Pestel, F. A. Leckie. *Matrix Method in Elastomechanics*. New York, San Francisco, Toronto, London: McGraw-Hill, 1963. 4, 31, 228
- [PW94] W. D. Pilkey, W. Wunderlich. *Mechanics of Structures. Variational and Computational Methods*. Boca Raton, Ann Arbor, London, Tokyo: CRC Press, 1994. 17, 18, 21, 31, 126, 225, 228

¹¹<http://books.google.ee/books?id=6tFZQT71Ff8C&pg=PA289&dq=Deformationsspr%C3%BCnge+Kr%C3%A4tzig&hl=en&sa=X&ei=TEEDUtuHKZHdsgbbv4D4BA&ved=0CC0Q6AEwAA#v=onepage&q=Deformationsspr%C3%BCnge%20Kr%C3%A4tzig&f=false>

¹²http://books.google.ee/books?id=ghco7svk5T4C&pg=PA3&lpg=PA3&dq=Andres+Lahe&source=bl&ots=3SFfo4UCES&sig=_XLUez-SfW2FVYGRx8v2LVm16V8&hl=et&ei=YQaFTMeIEoWc00yCyNwP&sa=X&oi=book_result&ct=result&resnum=5&ved=0CB0Q6AEwBDgK#v=onepage&q=Andres%20Lahe&f=false

¹³http://digi.lib.ttu.ee/opik_eme/Ehitusmehaanika.pdf

¹⁴http://www.ce.memphis.edu/3121/notes/notes_04d.pdf

¹⁵<http://www.ce.memphis.edu/3121/notes/notes.html>

¹⁶<http://www.ce.memphis.edu/3121/>

¹⁷<http://rutcor.rutgers.edu/~voudalov/lp2010handout1.pdf>

- [RH95] E. Ramm, T. J. Hofmann. Stabtragwerke. In: G. Melhorn (Hrsg), *Der Ingenieurbau: Baustatik und Baudynamik*. Berlin: Ernst & Sohn, 1995. S. 1–349. [126](#)
- [Rand07] K.D. Randall. *Physics for Scientists and Engineers. A Strategic Approach*. 2nd Edition.¹⁸ Pearson Addison-Wesley, 2007. Web. 08 August 2013. [29](#), [30](#), [40](#), [166](#), [222](#)
- [Suz00] R. M. De Souza. *Force-based Finite Element for Large Displacement Inelastic Analysis of Frames*. Fig. 2.2. Global and basic coordinates system.¹⁹ Berkely: University of California, Fall 2000. Web. 10 August 2013. [18](#)
- [Str59] В. В. Степанов. *Курс дифференциальных уравнений*. Москва: Государственное издательство физико-математической литературы, 1959. [136](#)
- [Str89] M. Stern. Static analysis of beams, plates and shells. In: D. E. Beskos (ed.), *Boundary Element Methods in Structural Analysis*. New York: Am. Soc. Civil Engineers, 1989, pp. 41–64. [4](#)
- [VrCty] Method of Sections. *Truss Structures*.²⁰ VirtualCity.²¹ Web. 09 August 2013. [223](#)
- [WP960] Beam on 2 Supports: Shear Force & Bending Moment Diagrams. Technical Description.²² GUNT Hamburg.²³ Web. 08 August 2013. [29](#), [30](#), [50](#), [132](#), [222](#), [223](#)
- [YSK08] L. L. Yaw, N.Sukumar, S. K. Kunnath. Meshfree co-rotational formulation for two-dimensional continua²⁴. *Int. J. Numer. Math. Engng*, 2008; 00:1-38. Web. 08 August 2013. [18](#)
- [Yaw09] L. L. Yaw. *2D Corotational Beam Formulation*.²⁵ Walla Walla University, November 30, 2009. Web. 27 August 2013. [52](#), [141](#)

¹⁸http://www.goodreads.com/author/show/10633.Randall_D_Knight

¹⁹http://www.ufpa.br/nicae/integrantes/remo_souza/TrabPublicados/TesesDissertacoesTccs/DissertationRemo.pdf

²⁰<http://vcity.ou.edu/demoModules/analysis/truss/truss.htm>

²¹<http://vcity.ou.edu/>

²²http://www.gunt.de/static/s3117_1.php?p1=&p2=&pN=;

²³http://www.gunt.de/static/s1_1.php?p1=&p2=&pN=

²⁴<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.216.9380&rep=rep1&type=pdf>

²⁵http://people.wallawalla.edu/~louie.yaw/Co-rotational_docs/2Dcorot_beam.pdf

Index

- active forces, 222
- area moment of inertia, 22
- axial displacement, 17
- axial force, 125
- axial force hinge, 29
- axial rigidity, 57

- basic coordinate system, 18, 52
- basic equations, 21
 - of beam, 110
 - of frame, 150
 - of truss element, 52
- basic stiffness, 31, 228
- beam governing equations, 25
- beam-column, 125, 130
 - effect, 125
 - element, 125
- boundary condition,
 - kinematic, 28
 - static, 28
- boundary problem equations, 60, 76, 86, 106, 152, 186
- boundary work, 172

- calculating trick, 50
- Cauchy formula, 23, 24, 136
- collapse load, 169
- collapse load factor, 207
- compatibility conditions, 39
- compatibility equations, 31, 34, 58, 74, 82, 148, 150, 157, 182, 201
- complete solution for a compressive axial force, 133
- computer function
 - ESTSKrmus.m, 245
 - ESTSTKrmus.m, 246
 - ESTtalaKrmus.m, 240, 245

 - InsertBtoA.m, 242
 - Lahe2FrameDFIm.m, 246
 - LaheBeamDFIm.m, 239
 - LaheFrame3hingeNQM.m, 237
 - LaheFrameDFIm.m, 235
 - LaheFrameSnDFIm.m, 251
 - LaheGerberBeamQM.m, 238
 - LaheTrussDFIm.m, 241
 - Sisejoud3LraamiPnktism.m, 238
 - SisejoudPunktism.m, 236
 - SisejoudTalaPunktis.m, 240
 - spInsertBtoA.m, 241
 - spSisestaArv.m, 242
 - SpToeSiirdeFiVektor.m, 242
 - SpToeSiirdeUvektor.m, 242
 - SpToeSiirdeWvektor.m, 242
 - SsjoudGrbrTalaPnktis.m, 239
 - yIffzII.m, 249
 - yIffhlin.m, 242
 - yIffhvzII.m, 248
 - yIffmhvII.m, 248
 - yIffmII.m, 249
 - yIffqII.m, 249
 - yIffqvII.m, 249
 - yIffplfhlin.m, 244
 - yIffplfmhvII.m, 248
 - yIffplvfmhvI.m, 244
 - yIffplvfmhvII.m, 250
 - yIffpSlfhlin.m, 244
 - yIffpSlvfmhvI.m, 245
 - yIffpSRhlin.m, 245
 - yIffpSRmhvI.m, 245
 - yIffpSTIffhlin.m, 244
 - yIffpSTIffvfmhvI.m, 245
 - yIffpTIffhlin.m, 244
 - yIffpTIffvfmhvI.m, 244

- yztransf.m, 250
- yztransfp.m, 250
- yzfzv.m, 243
- yzhqz.m, 242
- yzmyv.m, 243
- yzSfzv.m, 243
- yzShqz.m, 243
- yzSTfzv.m, 243
- yzSThqz.m, 243
- yzTfzv.m, 243
- computer program
 - spESTbeamLaheWFI.m, 239
 - spESTframe3hinge1NQM.m, 246
 - spESTframe3hinge1WFI.m, 246
 - spESTframe3hingeLaheNQM.m, 237
 - spESTframe3hingeLaheWFI.m, 237
 - spESTframe50WFI.m, 246
 - spESTframe77LaheWFI.m, 236
 - spESTframe93LaheWFI.m, 236
 - spESTframeLaheWFI.m, 235
 - spESTframeSn0LaheWFI.m, 252
 - spESTframeSn1LaheWFI.m, 252
 - spESTframeSn2LaheWFI.m, 252
 - spESTframeSn3LaheWFI.m, 251
 - spESTGerberBeamQM.m, 238
 - spESTGerberBeamWFI.m, 239
 - spESTtrussLaheWFI.m, 240
 - spESTtrussN2.m, 241
 - yspESTframe1LaheWFI.m, 248
 - yspESTframe2LaheWFI.m, 246
- connection,
 - contrived end, 26
 - fixed-support, 26
 - free end, 26
 - pin-support, 26
 - roller-support, 26
- contact forces, 29–31, 40, 222
- continuous beam, 82
- coordinates,
 - global, 32
 - local, 32
- DaF, 28
- delta function, 23
- direction cosines, 32, 219
- dissipation at plastic hinges, 207
- dissipation energy, 173
- elastic modulus, 22
- elastic section modulus, 171
- energy conservation, 223
- enlarged transfer matrix, 28, 227
- equilibrium at joint, 43
- equilibrium equation of beam, 212
- equilibrium equations, 31, 58, 148
- equilibrium of support node, 43
- equivalent distributed force, 23
- EST method, 194
- Euler-Bernoulli beam, 22
- external reactions, 26, 45
- external support, 31
- flexural rigidity, 57, 149
- force,
 - axial, 125
 - normal, 125
 - shear, 125
 - transverse shear, 125
- forces,
 - active, 222
 - contact, 29–31, 40, 222
 - reaction, 222
- frame, 73
 - three-hinged, 103
- full matrix, 216
- full plastic moment, 174, 177
- fundamental set of solutions, 132
- general solution of non-homogeneous differential equation, 130
- Gerber beam, 110, 211
- global coordinate system, 17, 47
- global coordinates, 32, 61, 218
- governing equations, 129
- Green's functional, 221
- Heaviside step function, 24
- hinge, 58
 - axial force, 29

- moment, 29
 - plastic, 46
 - shear force, 29
- homogeneous differential equation, 22
- ideal elastic-plastic, 169
- identity matrix, 220
- incremental method, 174
- influence line, 117
- influence line diagram, 120
- initial parameter vector, 49, 68, 76, 106, 186
- initial parameters, 23
- internal displacements, 18, 21
- internal reactions, 29, 30, 40, 45, 222
- internal support, 31
- inverse matrix, 220
- inverse transformation, 219, 220
- iteration, 147
- joint equilibrium equations, 40, 74, 82, 104, 110, 150, 157, 182, 201
- joint with,
 - axial force hinge, 29
 - free ends, 29
 - moment hinge, 29
 - shear force hinge, 29
- kinematic boundary condition, 28
- kinematic method, 174
- kinematic theorem, 173
- left-handed coordinates, 18
- limit design, 169
- load case, 89, 91, 181
- load factor, 169, 174, 175, 177
- load transformation, 103
- loading vectors, 141, 233
- local coordinates, 32, 218
- moment hinge, 29, 31, 150, 157
- nodal load, 189
- non-homogeneous differential equation, 23, 24
- normal force, 125
- normed fundamental set of solutions, 132
- orthogonal matrix, 220
- orthogonality, 128
- P- Δ effect, 125
- P-delta effect, 125
- particular solution, 23, 25
 - of the differential equation, 130, 136
 - with the Cauchy formula, 138
- plastic hinge, 46, 169, 171, 175
- plastic section modulus, 171
- plasticity, 169
- polygonal truss, 117
- rafter, 65
- rank of equations, 49
- reaction forces, 222
- restraints on support displacements, 26
- restrictions on support displacements, 47, 58, 74, 148, 157, 201
- right-handed coordinates, 17, 18
- rigid body
 - rotation, 17, 51
 - translation, 17, 51
- round-off error, 228
- scaling multiplier, 27, 31, 53, 144, 145, 228
- second-order analysis, 130
- set of fundamental solutions, 22
- shape factor, 171
- shear force, 125
 - diagram, 20
 - hinge, 29
- shear rigidity, 57
- side conditions, 31, 45, 58, 74, 104, 110, 148, 150, 157, 182, 201
- Sign Convention 1, 18, 23
- Sign Convention 2, 17, 18, 23, 40, 50
- sparse matrix, 211, 214–216
 - sparse(iv,jv,sv), 214
- sparsity pattern, 63, 95, 122
- state vector, 31
- static boundary condition, 28
- static equilibrium, 71

- statical method, 174
- superposition, 147
- support
 - displacements, 150
- support reactions, 87, 106, 201
- support shift, 47
- support,
 - contrived end, 26
 - external, 31
 - fixed, 26
 - internal, 31
 - pin, 26
 - roller, 26

- three-hinged frame, 103
- topology, 58, 110, 182
- transfer equations, 135, 228
- transfer matrix, 21, 25, 27, 61, 135, 137, 139, 182, 225
 - for a beam element with an axial force, 133
- transformation matrix, 32, 38, 143, 218
- transformation of a transfer matrix, 142
- transpose matrix, 220
- transverse shear force, 125
- truss, 96, 117
 - polygonal, 117
- truss element, 51
- two-span frame, 57, 157

- unit load method, 18

- work,
 - boundary forces, 222
 - external forces, 221
 - internal forces, 221
- Wronskian, 22, 23, 131