AFDELINGEN FOR

BÆRENDE KONSTRUKTIONER

DANMARKS TEKNISKE HØJSKOLE

STRUCTURAL RESEARCH LABORATORY

TECHNICAL UNIVERSITY OF DENMARK

Ole Holst

AUTOMATIC DESIGN OF PLANE FRAMES

RAPPORT NR. R 53 1974

Ole Holst

AUTOMATIC DESIGN OF PLANE FRAMES

Rapport nr. R 53 1974

Preface

This report is prepared in partial fulfilment of the requirements for the lic.techn. degree in Civil Engineering at the Technical University of Denmark.

Research leading to the present report was carried out at the Structural Research Laboratory from July 1971 to January 1974 under the supervision of Julius Solnes, lic.techn. and Mogens Peter Nielsen, dr.techn.

During the research years I have enjoyed valuable help from many of my colleagues at the laboratory. I particularly wish to thank Julius Solnes, lic.techn. and Esben Byskov, lic.techn. for advice and inspiration.

Ølstykke, November 1974

Ole Holst

Abstract.

In this report the minimum volume design of nonlinear geometric, linear elastic plane frames subjected to both constant and design dependent loads is examined and the computational procedures employed are reported in detail. Three design examples are thoroughly discussed.

It is concluded that the minimum volume design as reached through combined use of the time honoured fully stressed design concept and mathematical programming techniques involving sequential linear programming gives the designer a very efficient and economical way to design structures on a rational basis with a minimum of human intervention.

Table of contents

Notation.

A, AREA	Design variable, area.
A'	Auxillary design variable.
A_{max}	Maximum of A, fabricational limit.
A_{min}	Minimum of A, fabricational limit.
C	Shape factor in wind loading.
C	Structure factor in earthquake loading.
E, EMOD	Young's modulus.
F	Nodal force or load.
FI	Selected critical constraint value.
IAGRUP	Active groupnumber.
IAREST	Selected constraint number.
IBEL	Loadcase number
ICDIS	Displacement constraint number.
IDIS	Displacement number
IELM	Element number.
IPROF	Profile type number.
ITERA	Iteration number.
LAGRUP	List of active groups.
LBEL	List of IBEL's of selected constraints.
LCDIS	List of ICDIS'es of selected constraints.
LELM	List of IELM's of selected constraints.
LNCDIS	List of constrained displacements.
LNSNIT	List of NSNIT's
M	Moment in element.
M^o	Allowable M when N = O
MFI	Maximum constraint value.
N	Normal force in element.
N^o	Allowable N when M = O.

NAGRUP	Number of active groups
NAREST	Number of selected critical constraints.
NBEL	Number loadcases.
NCDIS	Number of controlled displacements.
NELM	Number of elements.
NFSI	Number of fully stressed iterations.
NGRUP	Number of groups.
NITERA	Maximum number of iterations.
NREST	Maximum number of selected critical constraints.
NSNIT	Number of stress check points in an element.
REDUKF	Reduction factor for movelimit.
RELAXF	Relaxation factor in fully stressed redesign.
S	Design volume.
T	Natural period.
UST	Maximum allowable displacement.
W,WMOM	Section modulus.
WM	Lumped mass
X	Displacement.

b_l	Lower bound on ΔA.
b_u	Upper bound on ΔA.
$f^{(ITERA)}$	Movelimit on $\Delta \underset{\sim}{A}$ in iteration ITERA
f_c	Allowable compressive stress factor.
f_p	Allowable proportionality stress factor.
g	Acceleration of gravity.
l	Length associated with A.
m	Fully stressed multiplier on A.
n_E	Factor of safety against Euler buckling.
r	Radius of gyration.

φ	Structural constraint value.
φ_s	Stress constraint value.
φ_d	Displacement constraint value.
γ	Specific gravity.
λ	Slenderness ratio.
λ_G	Transitional slenderness ratio
ψ	Merit function = S.
σ^B	Allowable stress in bending.
σ^C	Allowable stress in compression at $\lambda = 0$.
σ^N	Allowable stress in tension.
σ^P	Allowable proportionality stress.

Special notations.

(j = 1,M) means j = 1, 2, 3, ... M

$j \in$ LAGRUP means j = LAGRUP(1), LAGRUP(2), ... LAGRUP(NAGRUP)

$\sim$	vector or matrix
$\underset{\sim}{I}$, $\underset{\sim}{1}$	unit matrix, unit vector.
$\underset{\sim}{\nabla}$	gradient with respect to active design variables.
Δ	increment.

Introduction.

The subject of this report, automatic design of plane frames, is treated as a subproblem of the general optimal structural design problem which has received growing attention in the recent decades. A review of the general optimal design problem is given by Niordson & Pedersen in ref. [10].

Automatic design of frames with minimum volume using rigid plastic analysis is already a well established discipline since these design problems are linear in nature and thus both in theory and, after the advent of large scale digital computers also in practice, amenable to solution by direct linear programming techniques. For a review of plastic minimum volume design of frames see e.g. Livesley in ref. [12] containing many references to the developments within this field.

Minimum volume design of frames using linear elastic analysis is generally a nonlinear process and thus not quite as straightforward as rigid plastic design. The author's interest into this field was initially aroused by Romstad & Wang, ref. [11] containing many relevant references.

The present report augments the techniques laid down in ref.[11] in order to develop automatic structural design capabilities for even large, general plane frames with complex loadings.

Starting with the definition of a feasible structural design and a brief summary of pre-mathematical programming methods of obtaining a "good", feasible design the present report moves on to give the definition of structural design as an exercise in mathematical programming with minimum volume as the mathematical design criteria that ensures the automatic generation of a unique, feasible design of economical virtue.

The so-called sequential linear programming technique of solving the mathematical programming problem is formulated and the condition under which the fully stressed design has minimum volume is discussed in chapters 2 and 3.

In chapter 4 a recommended design strategy is presented and a flowchart of the general iterative design procedure is given. The content of the flowchart is elaborated in detail for a general plane frame with linear elastic behaviour in the remainder of chapter 4.

Chapter 5 thoroughly treats three design examples using different size plane rectangular house building frames subjected to several loadcases, including design dependent loadings. Great care has been taken in presenting realistic problems, using realistic loadings, using standard manufactures' profiles, imposing group design, using prestressed bracing elements, taking account of non-linear geometric analysis etc.

Based on the examples of chapter 5 and numerous other examples not reported here, a number of conclusions are put forth in chapter 6.

In appendix 1 is given a thorough descrip ion of the plane frame analysis which is employed in the developed design procedure.

Appendix 2 gives a description of the employed empirical relationships of moment of inertia and section modulus respectively to area for 3 standard manufactures' profiles and the theoretical relationships for the geometrically similar series.

Appendix 3 presents a generalized power method of solving an eigenproblem with a real symmetric matrix.

Finally, the derivations of chapter 4 and the first three appendices have been implemented in a modular computer program package which is listed in its entirety in appendix 4. The procedures are partly selfexplaining, partly explained in the above mentioned text.

1 Formulation of a feasible design.

A particular design of a frame is characterised by two sets of variables, the independent design variables, in this study chosen as the areas, and the dependent behavior variables like stresses, displacements and fundamental period.

Both the design- and the behavior variables generally are bounded by lower and/or upper limits.

Let $\underset{\sim}{\varphi}_s$ be the normalized stress vector, $\underset{\sim}{\varphi}_d$ the normalized displacement vector and $\underset{\sim}{A}$ the NGRUP dimensional design vector of areas. Then a feasible design is a design which obeys the following inequalities called the design constraints.

$$\underset{\sim}{1} \geq \underset{\sim}{\varphi}_s \qquad (1)$$

$$\underset{\sim}{1} \geq \underset{\sim}{\varphi}_d$$

and

$$\underset{\sim}{A}_{max} \geq \underset{\sim}{A} \geq \underset{\sim}{A}_{min}$$

This set of inequalities may have any number of solutions including none at all.

Structural frames used in housebuilding, however, always have infinitely many solutions when the elements in the design vector $\underset{\sim}{A}$ are regarded as continuous variables.

1.1 Feasible design by standard iterative methods.

Intuitively an economical design is one in which each design variable is at its lower limit or has been fully stressed under at least one loadcase. A feasible design which fulfills the above optimality criterion is hereafter called a fully stressed design. A reasonably fully stressed design may be reached after a number of iterations with individual scaling according to the stress level in the variables.

If constraints on the displacements are specified and any element in $\underset{\sim}{\varphi}_d$ is greater than 1 at the fully stressed design then the straightforward fully stressed redesign technique is not applicable to the problem.

To overcome this situation one may intuitively increase one or more variables that are thought most efficient in influencing the active displacement constraints or one may simply iteratively scale all variables with a common scale factor until $\underset{\sim}{\varphi}_d \leq \underset{\sim}{1}$. The first method intuitively includes economy in design but is ill defined and the second method, though well defined does not include any idea of economy in design.

Recently a novel method of redesign for active displacement constraints based on a so called optimality criterion has been reported by Gellatly & Berke in ref. [1]. This method promises a simple, well defined and optimal redesign method, but unfortunately some of the reported displacement constrained examples show unexplained instability of convergence.

2 Formulation of a minimum volume design.

The above mentioned design methods plus experience may lead to feasible designs of some economical merit.

A more rational approach, however, is to define a merit of the structure, $\psi = \psi(\underset{\sim}{A})$, and then design the structure through optimization of this merit while observing the design constraints (1).

The design problem is then cast as a mathematical programming problem which can be solved more or less automatically once the proper merit is chosen.

In this study the merit is a simple linear function of the design variables equal to the total material volume of the structure.

$$\psi = \underset{\sim}{A} \cdot \underset{\sim}{l} \rightarrow \text{minimum} \qquad (2)$$

Here $\underset{\sim}{l}$ is the vector of lengths associated with the design variables $\underset{\sim}{A}$.

While the merit (2) is a linear function of $\underset{\sim}{A}$ the set of constraints (1) normally are nonlinear functions of $\underset{\sim}{A}$ when the structure is statically indeterminate and non-linear analysis is performed.

Hence the mathematical programming problem belongs to the type known as bounded variable minimization of a linear function under a set of nonlinear constraints.

For this type of optimization many different algorithms have been proposed. See e.g. Fox, ref. [2] for a treatment of some of the algorithms. In this study only the so-called sequential linear programming technique (SLP) is treated.

2.1 Minimum volume design by sequential linear programming.

Had the constraints been linear the problem would have been an exercise in linear programming, (LP), which can be easily solved by means of the wellknown Simplex algorithm. See e.g. Holst, ref. [3] for one version of the algorithm.

When design constraints are nonlinear a straightforward approach is to approximate the constraints with first order Taylor series expansions involving computation of gradients of the constraints. Thereupon set up and solve the equivalent LP problem in the incremental design variables, add the increments to the current design and repeat the process until no improvement in the merit is possible.

The equivalent LP problem may be formulated as follows.

Merge $\underset{\sim}{\varphi}_s$ and $\underset{\sim}{\varphi}_d$ into $\underset{\sim}{\varphi}$, then the original nonlinear programming problem reads.

$$\min \psi = \underset{\sim}{A} \cdot \underset{\sim}{1}$$

provided

$$\underset{\sim}{1} \geq \underset{\sim}{\varphi}(\underset{\sim}{A}) \qquad (3)$$

$$\underset{\sim}{A}_{max} \geq \underset{\sim}{A} \geq \underset{\sim}{A}_{min}$$

After a change $\Delta\underset{\sim}{A}$ from the current design $\underset{\sim}{A}$, with $|\Delta\underset{\sim}{A}| \leq f \cdot \underset{\sim}{A}$ where f is intended to restrict the linearization error, the nonlinear problem is seen to be identical to the first order in $\Delta\underset{\sim}{A}$ with the following linear problem.

$$\min \Delta\psi = \Delta\underset{\sim}{A} \cdot \underset{\sim}{\nabla} \psi$$

provided

$$\underset{\sim}{1} - \underset{\sim}{\varphi} \geq \Delta\underset{\sim}{A} \cdot \underset{\sim\sim}{\nabla\varphi} \qquad (4)$$

$$\underset{\sim}{b}_u \geq \Delta\underset{\sim}{A} \cdot \underset{\sim\sim}{\nabla A} \geq \underset{\sim}{b}_l$$

where

$$\underset{\sim}{b}_u = \max(f \cdot \underset{\sim}{A}, \underset{\sim}{A}_{max} - \underset{\sim}{A})$$

$$\underset{\sim}{b}_l = \min(-f \cdot \underset{\sim}{A}, \underset{\sim}{A}_{min} - \underset{\sim}{A}) \qquad (5)$$

Application of the sequential linear programming technique using proper movelimit factors will always lead to a local minimum. If the feasible design space is convex then the local minimum is also the global minimum. In theory it is possible to obtain analytical expressions of the bounding constraints but in practice it is only done when the problems are very small and simple. Thus in general one cannot know the shape of the design space and thereby not know whether the found optimum is global or local.

The only way to minimize the probability that the found minimum is not global is by running a number of design processes with different initial design vectors.

2.2 Use of movelimits in SLP; strong and weak optima.

The factor f used in the upper and lower bounds on $\Delta \underset{\sim}{A}$ (5) called the movelimit was,originally introduced to restrict the linearization error inherent to the SLP technique.

Used in this way the proper value of f depends on the nonlinearity of the problem to be solved.

Generally problems that mostly depend on moments of inertia for strength and stiffness capabilities are more nonlinear than problems depending mainly on areas, i.e.,sway frame problems are more nonlinear than braced frame problems which in turn are more nonlinear than truss problems.

The more nonlinear the problems, the smaller the movelimit should be. Values between 0.1 and 0.5 are suggested. The final selection is a compromise between restricting the linearisation error and restricting the number of necessary iterations to convergence.

Using the movelimit factor exclusively to limit the linearization error only makes sense if the original nonlinear problem (3) has a strong optimum, that is, the minimum is determined by a NGRUP dimensional vertex in the NGRUP dimensional redesign space. If the original nonlinear problem has a weak optimum, that is, the minimum is determined by a vertex of dimension NAGRUP<NGRUP, then only NAGRUP independent constraints are active, i.e. have values 1.

This means that the minimum is flat in a NGRUP-NAGRUP+1 dimensional subspace. Stated in other words, at the minimum the merit function in this subspace is parallel to the tangent of one of the active constraint surfaces.

Then the variables corresponding to the subspace are poorly determined at the minimum. Roughly speaking, as long as these variables stay in the tangent plane the merit does'nt change and the constraint is only violated by a second order term in the deviations from the true minimum design.

Oscillations of the design variables after convergence of the merit is a sure sign of a weak optimum.

In order to force the variables to converge one method is to decrease the movelimit before every new iteration

$$f^{(ITERA+1)} = REDUKF \cdot f^{(ITERA)} \qquad (6)$$

The reduction factor and the initial movelimit are easily determined to yield a desired movelimit after a specific number of iterations. Values of REDUKF less than 0.6 should not be used. A value of REDUKF = 0.7 has shown the work well.

A second method to force convergence of the variables in presence of a weak optimum,more in agreement with the observation that the oscillating NGRUP-NAGRUP variables have small influence on merit and constraint values,is to assign these variables suitable intermediate values and then relegate them from further redesign. This could be accomplished by means of individual movelimit factors set to zero. However, a more efficient way is to divide all variables into an active set and an inactive set, the redesign then being executed only with respect to the active set of variables. This partition will be further described in a later section.

3 Fully stressed design versus minimum volume design.

Carrying out the design problem completely as a mathematical programming problem starting at a more or less arbitrary initial design is indeed possible but computationally very expensive in comparison with e.g. the simple fully stressed design iteration. For this reason it is well advised to take a look at the relationship between the fully stressed design and minimum volume design.

Using Lagrange multipliers Razani, ref. [4] and Kicher, ref. [5] have studied the relationship.

They conclude that for large classes of structures with 'normal action' the fully stressed design has minimum volume if stress constraints are the only constraints. Razani also observes that the faster the rate of convergence of the fully stressed design iterations, the more likely the optimality of the design.

Looking at the relationship from a design space point of view the following simple observations can be made.

If the minimum volume design is to be fully stressed in the extended sense then the minimum in the NGRUP dimensional design space must be a strong optimum, i.e., determined by the intersection of NGRUP stress or lower bound constraints each associated with a different variable.

Problems with stress constraints dominated by their associated variables, possibly with lower bounds, but without displacement constraints most likely have such strong optima and therefore constitute a class of problems with fully stressed minimum volume designs.

Both trusses and frames of low degree of statical indetermiracy and geometrical nonlinearity are candidates to this class. For this type of problems it is therefore a good idea to start the redesign iterations with a number of fully stressed iterations whereby the perhaps arbitrary initial design is vastly improved at low computational effort.

If the achieved fully stressed design actually is a minimum volume design this fact will be established after a few subsequent minimum volume redesigns.

If it is not minimum volume then usually several, costly minimum volume redesigns may be needed, were the algorithm allowed to run on until convergence of both merit and design variables.

4 Recommended design strategy.

Based on the foregoing remarks and experience gained from running a large number of test examples, the following 2 stage design strategy is recommended and has been used throughout the subsequent design examples.

Stage 1.

1 Make a crude estimate of the initial values of the design variables or simply use the maximum allowable values, $\underset{\sim}{A}_{max}$.

2 Use geometric linear analysis in steps 3-5.

3 Run 2 fully stressed iterations.

4 Run 1 iteration with a common scale factor based on the maximum constraint value after the fully stressed iterations. (Typically the maximum value belongs to a displacement constraint).

5 Run up to 5 minimum volume redesign iterations using SLP with decreasing movelimits.

Stage 2.

6 If stage 1 yields a weak optimum examine how many and which constraints are active at the current design. Decide which variables if any to relegate from further redesign. Assign appropriate manufactures'values to all variables.

7 If desired use,geometric nonlinear analysis and run the problem again with the perhaps reduced number of active design variables using only minimum volume redesign. If the correct set of active design variables has been chosen, only a few iterations will be necessary to achieve convergence to a strong optimum.

8 If the found optimal design looks abnormal then repeat steps 1 through 7 a number of times with different initial designs in order to minimize the possibility of having found a local minimum.

Both stages can be executed using the same iterative design procedure with different decision parameter values.

The iterative design procedure is well suited for flowchart representation.

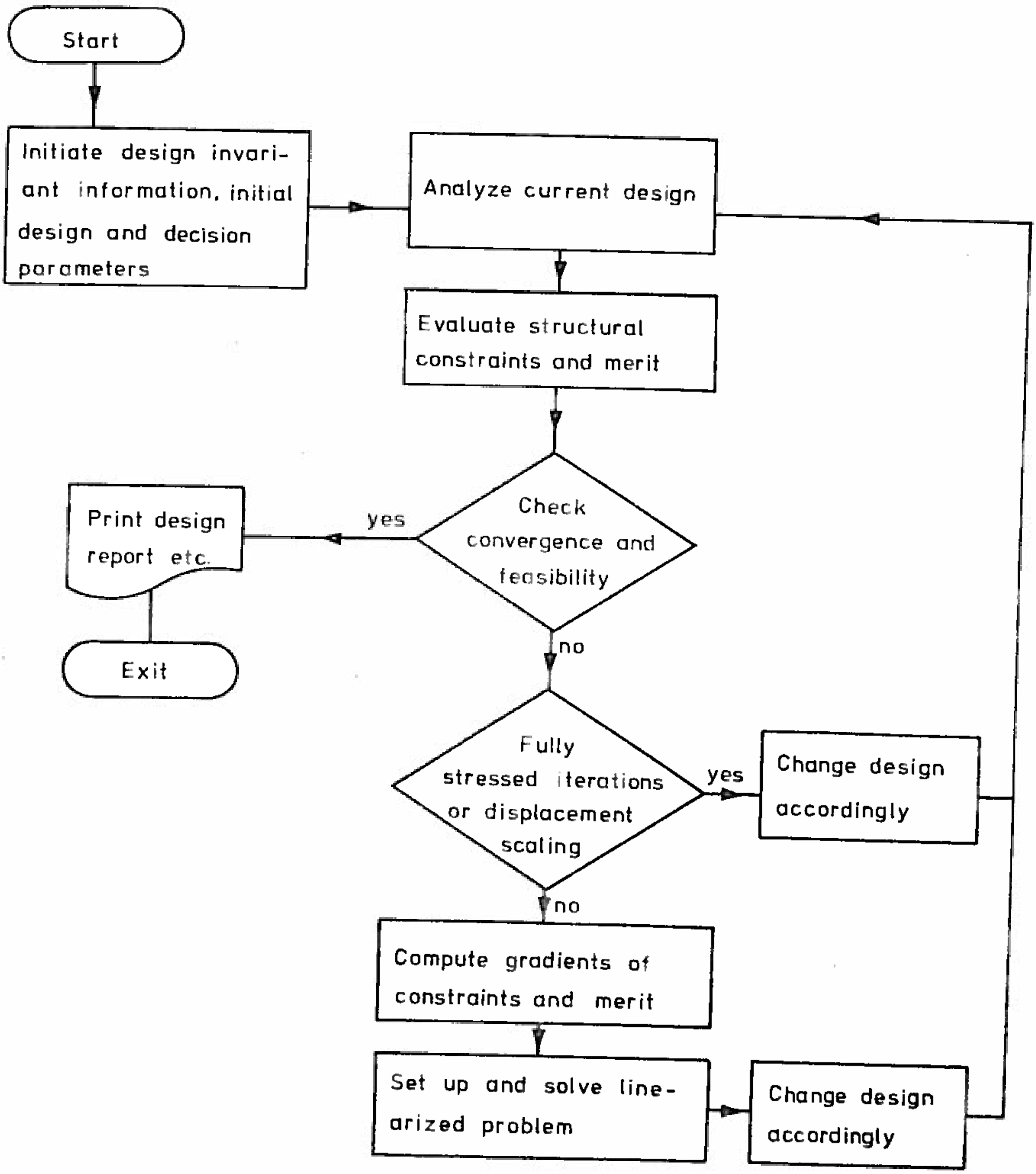


Fig. 1. Flowchart of iterative design procedure.

The content of the different boxes in the flowchart will be elucidated in the following chapters. References to the developed computer program are interspersed.

4.1 Initiate design invariant information; initial design and decision parameters.

The design invariant information consists partly of analysis information, partly of redesign information.

The analysis part is e.g. node coordinates, support conditions, element topology, element geometry, constant loads, type of analysis etc. as described in details in appendix 1. The redesign part is information about limits on allowable stresses, allowable displacements, allowable limits on the design variables, redesign stategy, active set of design variables etc.

All this information is either read in or generated once in procedure LÆS and is then kept as the invariant basis of the subsequent design computations.

To complete the description of a problem one has to specify the values of the design variables, i.e. the area vector $\underset{\sim}{A}$. (See appendix 1 for full description).

Good initial values of the design variables are really not crucial to the recommended design strategy. Even crude estimates of the area vector will suffice when a number of fully stressed iterations are performed before minimum volume redesign is attempted. If even crude estimates are not at hand one may simply start with the maximum allowable area vector.

Good estimates based on sound engineering judgement, however, never slows down a design process.

From the design vector $\underset{\sim}{A}$, the appropriate section properties are calculated by empirical or theoretical formulas as described in appendix 2.

Most decision parameters have default values assigned in the main procedure, DESIGN, but user's values may be read in through procedure LÆS.

4.2 Analyze current design.

The linear elastic analysis of the frame is carried out as described in appendix 1. It consists optionally of a linear geometric or nonlinear geometric analysis based on the displacement method. It is invoked by a call of procedure ANALYS.

The results of the analysis is the global displacement vector $\underset{\sim}{x}$ and optionally the fundamental period T of the frame. (T is used in the generation of equivalent earthquake loads). In principle also moments M and normal forces N result at NSNIT = LNSNIT (IELM) equidistantly spaced sections along each element, IELM, for all loadcases. M and N, however, are not stored but computed on basis of $\underset{\sim}{x}$ when needed in the stress constraint evaluation.

4.3 Evaluation of the merit and the structural constraints.

The merit is defined as the total steel volume used in the structure, i.e.

$$\text{merit} = \psi = \underset{\sim}{A} \cdot \underset{\sim}{1}$$

The evaluation of this expression presents no problem.

The structural constraints which are evaluated in procedure EVALFI consist of the stress constraints $\underset{\sim}{\varphi}_s$ and the displacement constraints $\underset{\sim}{\varphi}_d$.

4.3.1 Stress constraints.

The stress constraints are defined by the interaction formula

$$\varphi_s = \left|\frac{M}{M^o}\right| + \left|\frac{N}{N^o}\right| \leq 1 \quad (((\text{ISNIT} = 1,\text{NSNIT})\,\text{IELM} = 1,\text{NELM})\,\text{IBEL} = 1,\text{NBEL})$$
$$\text{NSNIT} = \text{LNSNIT(IELM)} \qquad (7)$$

Here the moment M and normal force N are computed as shown in appendix 1. Thus M and N are computed with due respect to geometric second order effects if nonlinear geometric analysis is requested.

The allowable moment M^o when N = O is defined by

$$M^o = W\sigma^B$$
$$= \text{WMOM(IGRUP)} * \text{SIGMAB(IBEL,IPROF)} \qquad (8)$$

where W is the section modulus and σ^B is the allowable stress in bending.

N^O, the allowable normal force when M = O,is defined by expressions depending on whether the element is in tension, $N \geq 0$, or compression $N < 0$.

In tension N^O is defined by

$$N^O = A\sigma^N$$

= AREA(IGRUP) * SIGMAN(IBEL,IPROF) (9)

where σ^N is the allowable stress in tension.

In compression, based on the slenderness ratio λ, a distinction is made between elements in the elastic or plastic range.

As a safe approximation the reduced lengths of both fix-fix and pin-pin type elements are defined equal to the geometric lengths, i.e. slenderness ratios are defined by

$$\lambda = \frac{l}{r} \tag{10}$$

where

$$r = \sqrt{\frac{I}{A}} = \text{radius of gyration} \tag{11}$$

The following definition of the allowable normal force in compression is based on a proposal by Pauli Pedersen in ref. [8].

Define

$\sigma^C = f_C\sigma^N$ = allowable stress in compression at $\lambda = 0$

n_E = factor of safety against Euler buckling.

$\sigma^P = n_E f_P \sigma^N$ = allowable proportionality stress

then the transition from the plastic region to the elastic region is determined by the transitional slenderness ratio λ_G.

$$\lambda_G = \pi \sqrt{\frac{E}{\sigma^P}} \tag{12}$$

In the plastic region, $\lambda \leq \lambda_G$, the allowable normal force is then given by

$$N^O = A(\sigma^C(1-(\frac{\lambda}{\lambda_G})^2) + \sigma^P(\frac{\lambda}{\lambda_G})^2/n_E)$$

$$= A\sigma^N(f_C-(f_C-f_P)(\frac{\lambda}{\lambda_G})^2) \qquad (13)$$

In the elastic region, $\lambda > \lambda_G$ the allowable normal force is given by

$$N^O = A\sigma^P(\frac{\lambda}{\lambda_G})^{-2}/n_E$$

$$= A\sigma^N f_P(\frac{\lambda}{\lambda_G})^{-2} \qquad (14)$$

As shown by Pauli Pedersen in ref. [8] these expressions are continous at $\lambda = \lambda_G$ regardless of the values of f_C and f_P. If we choose $f_C = 2f_P$ then even the first derivatives will be continuous at $\lambda = \lambda_G$. Default values used in the program and in the design examples in chapter 5 are

$$f_C = 0.8$$
$$f_P = 0.4 \qquad (15)$$
$$n_E = 1.7$$

Other values may be specified on input.

Using the default values one finds the following expressions

$$\lambda \leq \lambda_G : N^O = 0.8A\sigma^N(1-\tfrac{1}{2}(\frac{\lambda}{\lambda_G})^2)$$
$$\lambda > \lambda_G : N^O = 0.4A\sigma^N(\frac{\lambda}{\lambda_G})^{-2} \qquad (16)$$

4.3.2 Displacement constraints.

Although one may easily treat constraints on linear combinations of nodal displacements in order to include e.g. storey drift limits, only constraints directly on nodal displacements are incorporated in the program.

A total of NCDIS displacement directions are constrained.

Let UST(IBEL,IDIS) (>0) equal the maximum allowable displacement in loadcase IBEL and nodal direction IDIS, then the normalized displacement constraints are

$$\varphi_d = \frac{|X(IBEL,IDIS)|}{UST(IBEL,IDIS)} \leq 1 \quad ((IDIS \in LNCDIS)IBEL = 1,NBEL) \quad (17)$$

As indicated the global displacement direction IDIS in which the displacement is constrained is given by the topological array LNCDIS where

$$IDIS = LNCDIS(IBEL,ICDIS) \quad (ICDIS = 1,NCDIS) \quad (18)$$

If any element in LNCDIS is equal to zero the associated constraint is undefined and ignored. (This situation arises when a displacement is constrained only in some of the loadcases).

4.3.3 Automatic selection of a set of critical constraints.

The total number of stress and displacement constraints easily grows very large.

Bearing in mind that one must find the gradients of the constraints in order to perform the minimum volume redesign by sequential linear programming one naturally wishes to treat only those constraints which are active in the solution to the linear programming problem.

It is not possible,however,to foresee exactly which constraints will be active in the final design, but the following selection procedure drastically lowers the number of constraints to be considered without excluding potentially active constraints.

The maximum number of structural constraints selected is NREST = NGRUP + NBEL * (NGRUP + NCDIS). The actual number selected, called NAREST, may be smaller than NREST, thus further diminishing the size of the linear programming problem. The selected constraint values are placed in the array FI.

In order to ensure inclusion of at least 1 stress constraint per design variable and to facilitate ease in eventual fully stressed iterations, the first NGRUP design constraints are always selected as the largest stress constraint values within each group, i.e. FI(IGRUP) contains the maximum value of φ_s obtained at any control section within the elements belonging to design variable IGRUP under any loadcase.

In addition to these NGRUP stress constraints a number of other critical constraints are selected. These additional critical constraints are structural constraints whose value exceed $1-\varepsilon$, where ε is chosen reasonably small but large enough to avoid oscilliatory behavior of the sequential linear programming process. A value of $\varepsilon = 0.2$ has shown to be reasonably discriminatory without being too small. In case of more than NREST-NGRUP additional critical constraints only the NREST-NGRUP largest ones are selected.

4.3.4 Constraint identification.

In order to identify the constraints in FI a number of pointer arrays are used. These are defined as follows.

If LCDIS(IAREST) = 0 then the constraint is a stress constraint within element LELM(IAREST) with the dimensionless distance of the critical control section from the start of the element given by LKSI(IAREST).

If LCDIS(IAREST) > 0 then the constraint is a displacement constraint identified by ICDIS = LCDIS(IAREST).

The pertinent loadcase no. in both cases is given by LBEL(IAREST).

Out of the NAREST selected critical design constraints, the maximum value of φ, called φ_{max} = FIMAX = MFI, is found and stored for later use in the feasibility check and the possible displacement scaling.

The above explained automatic critical constraint selection and idenfication plus the φ_{max} calculation is performed in the procedure EVALFI.

4.4 Convergence and feasibility check.

The design process is said to have converged if the relative change in total volume during the iteration fall below a certain value; in the program 0.1%.

Convergence criterion: $\Delta\underset{\sim}{A}\cdot\underset{\sim}{1} < \underset{\sim}{A}\cdot\underset{\sim}{1}/1000$ (19)

Note that this convergence criterion does not guarantee convergence of the individual elements in $\underset{\sim}{A}$. In case of a weak optimum oscillations in the elements are possible even when the convergence criterion is satisfied.

A design is said to be feasible if no constraints are violated, i.e. if $\varphi_{max} \leq 1$.

Because the solution to the nonlinear minimization problem is obtained through a sequence of linear approximations a less strict feasibility criterion than the theoretical is used.

Feasibility criterion: $\varphi_{max} \leq 1.002$ (20)

When both criteria are satisfied or a specified maximum number of iterations, NITERA, has been reached the design process is stopped and a design report is printed. Optionally stresses and displacements of the final design may be printed and a drawing of the displaced structure may be plotted.

4.5 Redesign methods; active versus inactive variables.

As earlier noted three different redesign methods are included in the program. Relaxed fully stressed iterations with individual scaling of design variables, scaling for φ_{max} (normally associated to a displacement constraint) with a common scale factor, and minimum volume redesign by sequential linear programming.

An important common feature to all these redesign methods is the partitioning of the design variable vector into a set of active and a set of inactive variables. Only the active variables are eligible for redesign, thus making it possible to keep the set of inactive variables at fixed values during the design process. The number of active variables is NAGRUP $\leq$ NGRUP, and the connections between active group numbers and original group numbers are given by

IGRUP = LAGRUP(IAGRUP) (IAGRUP = 1, NAGRUP) (21)

In the description of the three different redesign methods it will be tacitly understood that the design and incremental design variable vectors, $\underset{\sim}{A}$ and $\Delta\underset{\sim}{A}$, now only consist of the active design variables.

Redesign only with active variables does not alter the number of structural constraints, but the dimensionality of the redesign space will be NAGRUP rather than NGRUP. This constitutes important savings in computation time when gradients of structural constraints are to be evaluated for use in the minimum volume redesign method.

4.5.1 Relaxed fully stressed iterations.

In the program NFSI-1 fully stressed iterations will be performed initially if NFSI > 0.

In fully stressed iterations all design variables that are understressed, i.e. have $\varphi_s < 1$, are decreased and variables that are overstressed are increased.

The iterations are simple, calling for just 1 analysis per iteration. The incremental design variables are simply given by

$$\Delta A_j = m_j A_j \qquad (j \in \text{LAGRUP}) \qquad (22)$$

where m_j is a multiplier performing the mentioned increase or decrease.

A straightforward approach is to define

$$m_j = \varphi_{sj} - 1 \qquad (j \in \text{LAGRUP}) \qquad (23)$$

where φ_{sj} is the maximum normalized stress constraint belonging to group j as defined in section 4.3.3.

This definition of m_j is qualitatively correct but may lead to oscillations. A relaxed version therefore has been used

$$m_j = \text{RELAXF}\ (\varphi_{sj} - 1) \qquad (j \in \text{LAGRUP}) \qquad (24)$$

The correct value of RELAXF is influenced by 2 factors, each pulling in opposite directions. Stiffening a member in an indeterminate structure yields a higher stress level in the member and conversely with weakening a member. This calls for overrelaxation, i.e. RELAXF > 1. In frames however one must also take into account that moment of inertia vary much more rapidly than area which calls for underrelaxation, i.e. RELAXF < 1. In frames the second factor is much stronger than the first and underrelaxation must be performed in order to avoid heavy oscillations in the fully stressed iterations. A value of RELAXF = 0.7 has worked well in several frame examples and this value is used in all the subsequent design examples.

In trusses a small overrelaxation is justified but since divergence easily may result if RELAXF is chosen too large it is not recommended to overrelax even if it may speed up convergence.

In the extended definition of a fully stressed design, variables are also permitted to be on their minimum or maximum allowable values and the incremental design variables are then given by

$$\begin{aligned} m_j > 0: \quad \Delta A_j &= \min(m_j A_j, A_{max,j} - A_j) \\ & \qquad\qquad (j \in \text{LAGRUP}) \qquad (25) \\ m_j \leq 0: \quad \Delta A_j &= \max(m_j A_j, A_{min,j} - A_j) \end{aligned}$$

In relaxed fully stressed iterations convergence is usually rapid and only a few iterations are necessary to furnish a reasonably fully stressed design.

4.5.2 Scaling for displacement constraints.

After performing the fully stressed iterations one may still have $\varphi_{max} > 1$. Most likely this is because one or more displacement constraints are violated at the fully stressed design. Iteration number NFSI (if NFSI > 0) is therefore a scaling of all active design variables with a common scale factor f.

$$\Delta \underset{\sim}{A} = \min(f\underset{\sim}{A}, \underset{\sim}{A}_{max} - \underset{\sim}{A}) \qquad (26)$$

Displacements are stiffness dependent and for trusses where stiffness is proportional to the areas a good strategy is to scale with

$$f = \varphi_{max} - 1 \qquad (27)$$

In frames, particularly sway frames, stiffness also depends on moments of inertia, which in most profile series are power functions of area with an exponent of at least 2. Hence a good strategy is to use

$$f = \sqrt{\varphi_{max}} - 1 \qquad (28)$$

The latter scale factor is used in the subsequent design examples.

4.5.3 Minimum volume redesign using sequential linear programming.

The formulation in section 2.1 of the linear programming problem to be solved at the current design is not altered when redefining $\Delta \underset{\sim}{A}$ to consist of only the set of active variables, i.e. formulating the redesign problem in a NAGRUP dimensional active redesign sub-space of the NGRUP dimensional total design space.

4.5.3.1 Gradient computations.

As seen from equation (4) gradients of both merit and structural constraints are needed when setting up the sequential linear programming problem.

The gradient of the merit is easily calculated because the merit ψ is a linear function in the active design variables.

$$\underset{\sim}{\nabla}\psi = \underset{\sim}{\nabla}(\underset{\sim}{A}\cdot\underset{\sim}{l}) = \underset{\sim}{l} \tag{29}$$

where $\underset{\sim}{l}$ is the vector of lengths associated with the active design variables.

The gradients of the structural constraints, however, are not so quickly calculated but require rather time consuming computations whether they are calculated analytically or numerically. This is the main reason why one should try to keep down the number of active design variables when performing redesign with gradient dependent methods.

As indicated by Solnes & Holst in ref. [9] it is perfectly possible to obtain the structural gradients analytically, but for problems involving nonlinear geometric analysis of general plane frames with design dependent loads the programming involved becomes too lengthy and complicated. For this reason and also because of greater ease in treating new design dependent loads a numerical differentiation has been preferred.

Both a simple forward and a simple backward difference scheme has been tested. Either approach seem equally well suited, i.e. give the same design in test examples. The present version of the program uses the simple backward difference scheme with a - 1% change in the differentiating variable.

The numerical differentiation requires NAGRUP reanalyses and NAGRUP *·NAREST constraint evaluations.

Reanalysis is performed with the same procedure, ANALYS, used to analyze the current design except that the nonlinear geometric analysis equations are only iterated once because experience has shown that the solution converges in one iteration when the change in design is a mere 1% change of one variable.

The constraint evaluation is also done by the same procedure, EVALFI, used to evaluate the current design. However, only the selected set of critical constraints contained in the array FI and identified by the pointer arrays of section 4.3.4 are evaluated instead of the total set of constraints.

Upon completing the numerical differentiation of the structural constraints one has the desired values of $\nabla\underset{\sim}{\varphi}$.

4.5.3.2 Setting up and solving the linear programming problem.

In order to solve the LP problem

$$\begin{aligned} &\text{minimize} && \Delta\psi = \Delta\underset{\sim}{A}\cdot\underset{\sim}{\nabla}\psi \\ &\text{provided} && \underset{\sim}{1}-\underset{\sim}{\varphi} \geq \Delta\underset{\sim}{A}\cdot\underset{\sim}{\nabla\varphi} \\ &\text{and} && \underset{\sim}{b}_u \geq \Delta\underset{\sim}{A}\cdot\underset{\sim}{I} \geq \underset{\sim}{b}_l \end{aligned} \qquad (4)$$

one must transform it into the standard LP form, i.e. state it in nonnegative variables.

This is accomplished by a parallel transformation of the redesign variables $\Delta\underset{\sim}{A}$.

Define the auxillary redesign variables $\Delta\underset{\sim}{A}'$, where

$$\Delta\underset{\sim}{A}' = \Delta\underset{\sim}{A}-\underset{\sim}{b}_l \qquad (30)$$

then the above LP problem is identical to the following auxillary LP problem in standard formulation ($\Delta\underset{\sim}{A}' \geq \underset{\sim}{0}$)

$$\begin{aligned} &\text{minimize} && \Delta\psi' = \Delta\underset{\sim}{A}'\cdot\underset{\sim}{\nabla}\psi \\ &\text{provided} && \underset{\sim}{1}-\underset{\sim}{\varphi}-\underset{\sim}{b}_l\cdot\underset{\sim}{\nabla\varphi} \geq \Delta\underset{\sim}{A}'\cdot\underset{\sim}{\nabla\varphi} \\ &\text{and} && \underset{\sim}{b}_u-\underset{\sim}{b}_l \geq \Delta\underset{\sim}{A}'\cdot\underset{\sim}{I} \end{aligned} \qquad (31)$$

The auxillary LP problem is characterized by positive coefficients of the merit function and (normally) more constraints than variables, which makes the problem ideally suited for solution by a dual simplex strategy (cfr. Holst, ref. [3]). Hence the dual strategy as implemented in procedure DUALP is used in the program.

After solution of the auxillary LP problem the true redesign vector is given by

$$\Delta \underset{\sim}{A} = \Delta \underset{\sim}{A}' + \underset{\sim}{b}_l \tag{32}$$

If the current design is infeasible and the bounds on $\Delta \underset{\sim}{A}$ do not permit large enough moves to satisfy all the linearized structural constraints or if the linearized structural constraints define an empty solution set then the LP problem has no solution. In this case the redesign vector is increased by the upper bounds

$$\Delta \underset{\sim}{A} = \underset{\sim}{b}_u \tag{33}$$

and a new minimum volume redesign cycle is attempted.

5 Design examples.

Though the developed computer program is capable of designing plane frames of arbitrary shape, only rectangular house building frames have been studied in detail.

The three frames presented in this chapter are examples of low, medium and high rise frames. All frames are designed against vertical and horizontal loads and horizontal displacement constraints play a major role in the design of some of the frames. Regarding the displacement control it is interesting to notice the difference in the distribution of added stiffness from medium to high rise frames.

As representatives from the three classes of frames a 2 storey 1 bay, a 10 storey 3 bay and a 30 storey 2 bay frame is elastic minimum volume designed using the recommended design strategy of chapter 4, including nonlinear geometric analysis in stage 2. Stress constraints are evaluated at element ends and midpoints.

For brevity only one initial design is used for each example.

For each design a design report containing the iteration history of all design variables, $\underset{\sim}{A}$, the total steel volume, S, the most critical constraint value, φ_{max} = MFI, and possibly the natural period, T, is presented both in tabular and in graphical form.

In the tabular design reports, areas designed from fully stressed or common scaling, and areas designed on movelimits rather than structural constraints are prefixed by a minus sign.

In the graphical representation areas designed on movelimits are singled out by a solid dot.

Also shown for each design is a plot of the largest horizontal displacements with the physical occurrences of the critical stress constraints. Fully stressed constraints are marked with a solid dot, while non fully stressed critical constraints are marked with a circle and the constraint value. Displacement constraints are graphically indicated.

All these devices are helpful in determining which areas should possibly be fixed in stage 2 of the recommended design strategy. The plot with the critical constraint values also gives a good picture of which parts of the structure are displacement controlled and which are stress controlled.

5.1 Example 1, 2 storey 1 bay.

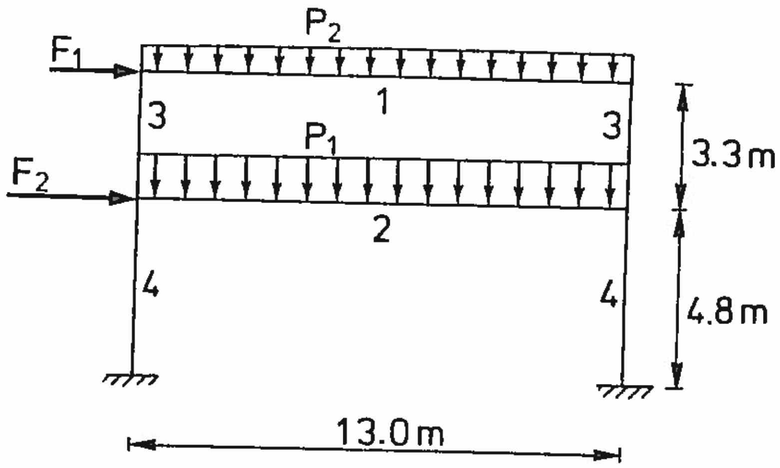


Fig. 2 Frame OH-2E1F, loadings, group numbers and geometry.

The loads are

$P_1 = 0.7$ Mp/m
$P_2 = 1.4$ Mp/m
$F_1 = 2.5$ Mp
$F_2 = 7.5$ Mp

The allowable stresses in tension and bending are $\sigma^N=\sigma^B=2\cdot 10^4$ Mp/m^2. The 6 elements are grouped into 4 groups as shown on fig. 2. The beams are IPE profiles and the columns are HEB profiles.

5.1.1 Plastic design.

In ref. [3] the author has designed this frame noniteratively as a minimum volume plastic design using the equilibrium method of limit design and linear programming. In the following table the resulting plastic design moments, the nescessary profiles and the associated areas are shown.

Group no.	Design moment (Mpm)	Nescessary profile	Area (cm^2)
1	9.12	IPE 300	53.8
2	21.12	IPE 400	84.5
3	9.12	HEB 200	78.1
4	12.00	HEB 220	91.0

Table 1. Discretized optimal plastic design of frame OH-2E1F.

The total steel volume of this plastic design is 0.3186 m^3.

5.1.2 Elastic design.

The same problem, named OH-2E1F was run as an elastic minimum volume design problem.
The graphic design report fig. 3 shows a fully stressed minimum volume design with a total steel volume of 0.3560 m^3 based on continous variables. The discrete design with S = 0.3631 and 3% overstress is used as the initial design for stage 2. The differences in designs from stage 1 and stage 2 are very small which implies that nonlinear geometric effects are negligible for this type of low rise building frames.
From the graphic design report is also noted that the convergence is rapid to a strong optimum.

The discrete design is shown below

Group no.	Optimal design area (cm^2)	Discrete profile	Profile area (cm^2)
1	58.8	IPE 330	62.6
2	95.65	IPE 450	98.8
3	79.23	HEB 200	78.1
4	109.44	HEB 240.	106.0

Table 2. Discretized optimal design of frame OH-2E1F.

5.1.3 Elastic design with displacement constraints.

Both previous designs of the 2 storey 1 bay frame have relative displacements well in excess of 1/500 which is the maximally permitted by the present Danish steel code.

Thus neither designs are acceptable and the elastic design is run again, this time including relative displacement constraints of 1/500 on the horizontal displacement of the top left corner and on the vertical displacements of the midpoints of both beams.

In order to make this second elastic minimum volume design as realistic as possible, 3 loadcases rather 1 are considered. Load-case no. 1 is identical to the original loadcase while loadcase

OH-2E1F NBEL=1
ISTAB= 0 NAGRUP= 4 NFSI= 3 RELAXF= 0.70 REDUKF= 0.70 FFM= 0.40 UNITS=MP,M

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2

	0	1	2	3	4	5
1	100.00	-60.61	-58.36	-58.45	58.30	58.30
2	100.00	-93.06	-95.42	-95.57	95.26	95.26
3	100.00	-79.34	-78.92	-79.04	78.55	78.55
4	100.00	-103.78	-108.45	-108.62	108.83	108.84
S	0.4220	0.3518	0.3561	0.3567	0.3560	0.3560
MFI	1.0540	1.0642	1.0032	1.0006	1.0000	1.0000

Table 3. Design report stage 1.

OH-2E1F NBEL=1
ISTAB= 5 NAGRUP= 4 NFSI= 0 RELAXF= 0.70 REDUKF= 0.70 FFM= 0.40 UNITS=MP,M

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2

	0	1	2	3
1	62.60	58.60	58.80	58.80
2	98.80	95.57	95.66	95.65
3	78.10	79.34	79.24	79.23
4	106.00	108.11	109.43	109.44
S	0.3631	0.3566	0.3581	0.3581
MFI	1.0332	1.0151	1.0001	1.0000

Table 4. Design report stage 2.

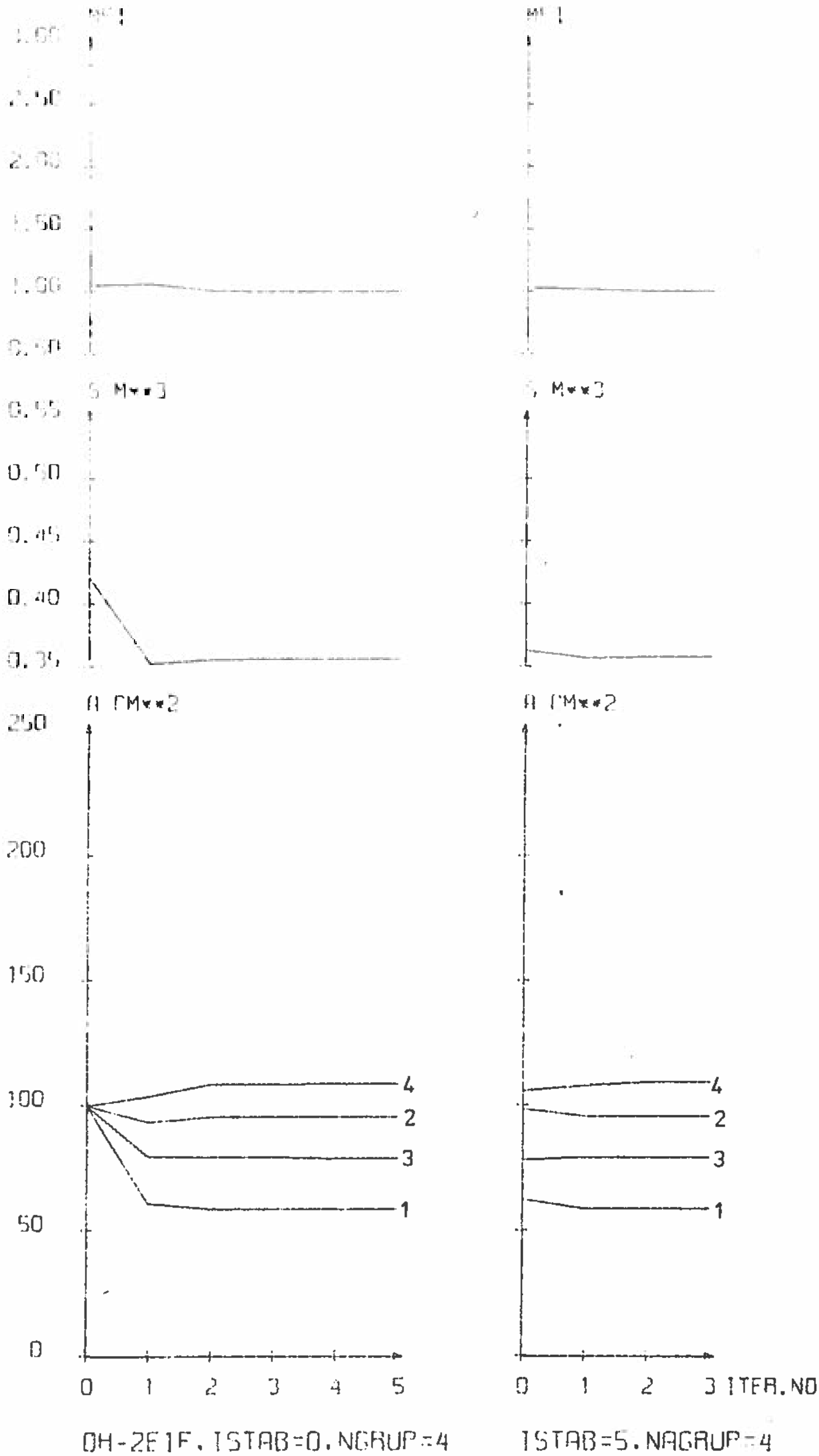


Fig. 3. Graphic design report, stage 1 and stage 2.

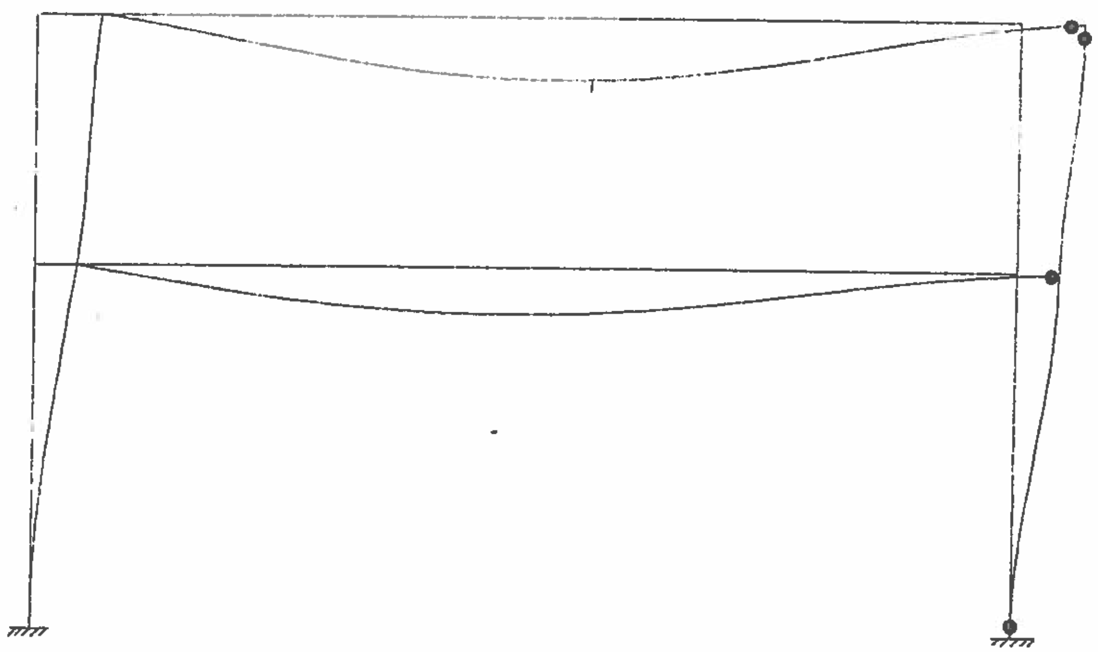


Fig. 4. Frame OH-2E1F, most critical values of φ_s within each group. Displacements x20.

no. 2 consists of the horizontal loads alone and no. 3 of the vertical loads alone. Also selfweight of the elements is included in all loadcases. This design problem is named OH-2E1FU.

As illustrated by the graphic design report fig. 5 the redesign path is quite different from that of OH-2E1F shown on fig. 3. After 2 convergent fully stressed iterations φ_{max} = MFI = 2.6 (associated with one of the displacement constraints). The common scaling for MFI > 1 given by equation (28) brings MFI below 1 but overestimates the necessary steel volume. The subsequent 5 minimum volume redesigns show that the lower column, design variable no. 4, plays a major role in the minimum volume displacement control.

The convergence of the steel volume, S, is stable and so is the convergence of the 2 column variables, design variables no. 2 and no. 4. Design variables no. 1 and no. 3, the 2 beams, however, show oscillating behavior. An examination of all critical constraints after the termination of stage 1 reveals that only 1 stress constraint, belonging to variable no. 2, and 2 displacement constraints are active. With a total of 3 active constraints and 4 active variables we have a weak optimum and 1 of the 2 oscillating design variables must be deactivated in the stage 2 run. Exhibiting the larger oscillations, variable no. 3 is chosen for deactivation. The 2 active displacement constraints are the one in the upper left corner and the one in the midpoint of the upper beam.

The discrete design used as initial design in stage 2 has a steel volume of S = 0.5022 m^3 with a 4% excessive displacement (MFI=1.04). The convergence in stage 2 is rapid to a strong optimum.

The optimal design with due respect to displacement limits thus uses 42% more steel than the optimal design without displacement limits.

Running stage 2 both with and without nonlinear geometric analysis showed only neglible differences.

The discrete design is shown below in table 5.

Group no.	Optimal design area (cm^2)	Discrete profile	Profile area (cm^2)
1	74.59	IPE 360	72.7
2	91.80	IPE 450	98.8
3	106.00	HEB 240	106
4	219.69	HEB 450	218

Table 5. Discretized optimal design of frame OH-2E1FU.

```
OH-2E1FU NBEL=3 MAX U/L=1/500  NCDIS=7                                 UNITS:MP,M
 ISTAB= 0 NAGRUP= 4 NFSI= 3 RELAXF= 0.70 REDUKF= 0.60 FFM= 0.40

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2
```

	0	1	2	3	4	5	6	7	8
1	100.00	-62.25	-75.[illegible]	-95.82	-57.40	67.57	-77.70	73.7[illegible]	73.33
2	100.00	-95.35	-96.97	-156.00	-93.60	92.20	91.85	91.[illegible]	91.90
3	100.00	-82.05	-81.72	-131.75	-73.05	98.02	-52.61	-1[illegible].01	-[illegible].39
4	100.00	-105.21	-108.94	-175.63	205.10	217.30	219.50	219.32	21[illegible].93
S	0.4220	0.3607	0.3618	0.5829	0.4455	0.4810	0.4918	0.49[illegible]	0.4944
MFI	2.5118	2.7079	2.5291	0.8[illegible]38	1.8355	1.2223	1.0502	1.0017	1.0043

Table 6. Design report, stage 1.

```
OH-2E1FU NBEL=3 MAX U/L=1/500  NCDIS=3                                 UNITS:MP,M
 ISTAB= 5 NAGRUP= 3 NFSI= 0 RELAXF= 0.70 REDUKF= 0.60 FFM= 0.40

DESIGN REPORT
DOWN:GROUPNO.,AREA IN CM**2,RIGHT: ITERATIONNO.
```

	0	1	2
1	72.70	74.70	74.59
2	98.80	91.25	91.80
3	106.00	-106.00	-106.00
4	218.00	220.32	219.69
S	0.5022	0.4972	0.4972
MFI	1.0410	1.0076	1.0003

Table 7. Design report, stage 2.

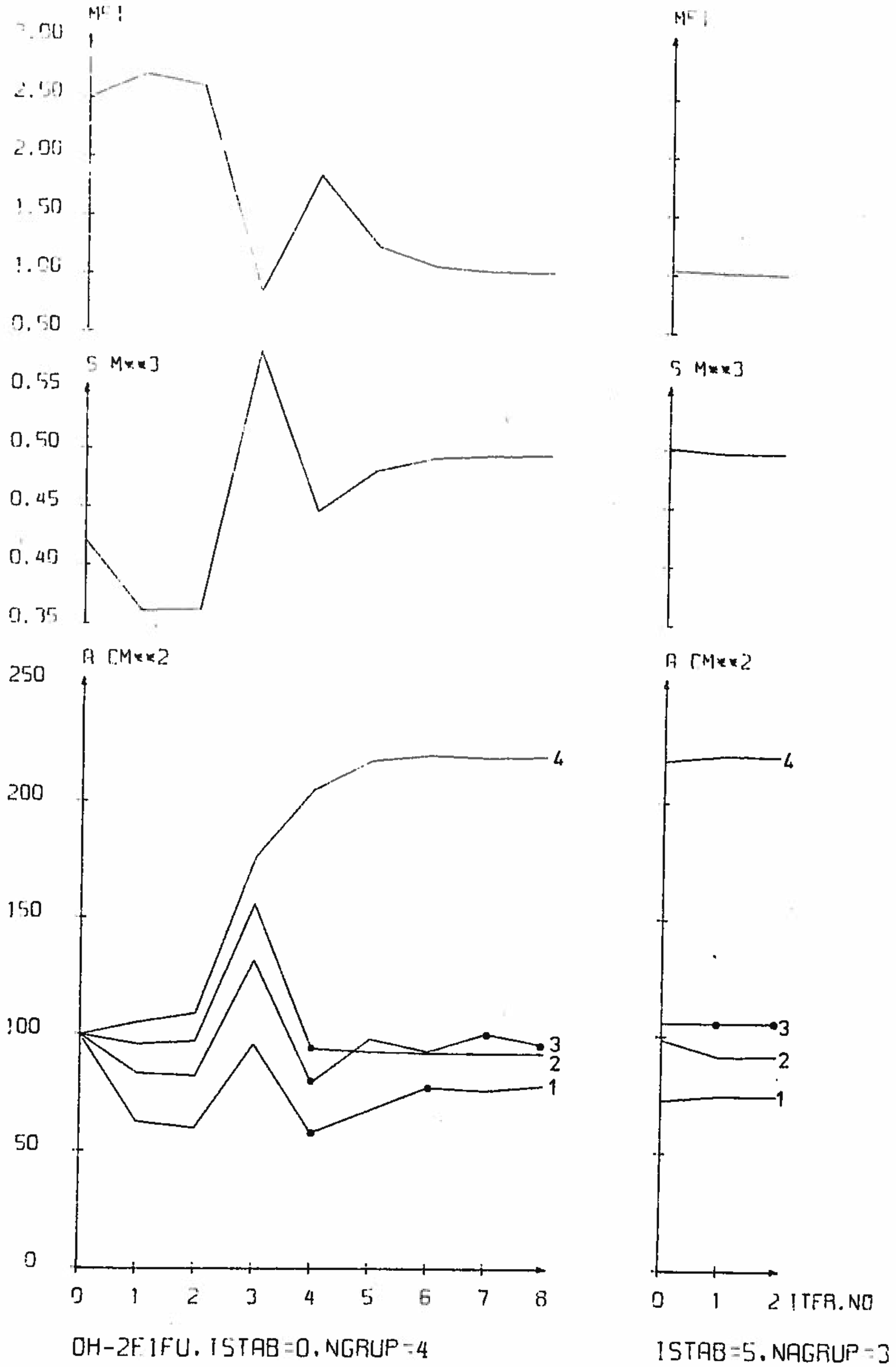


Fig. 5. Graphic design report, stage 1 and stage 2.

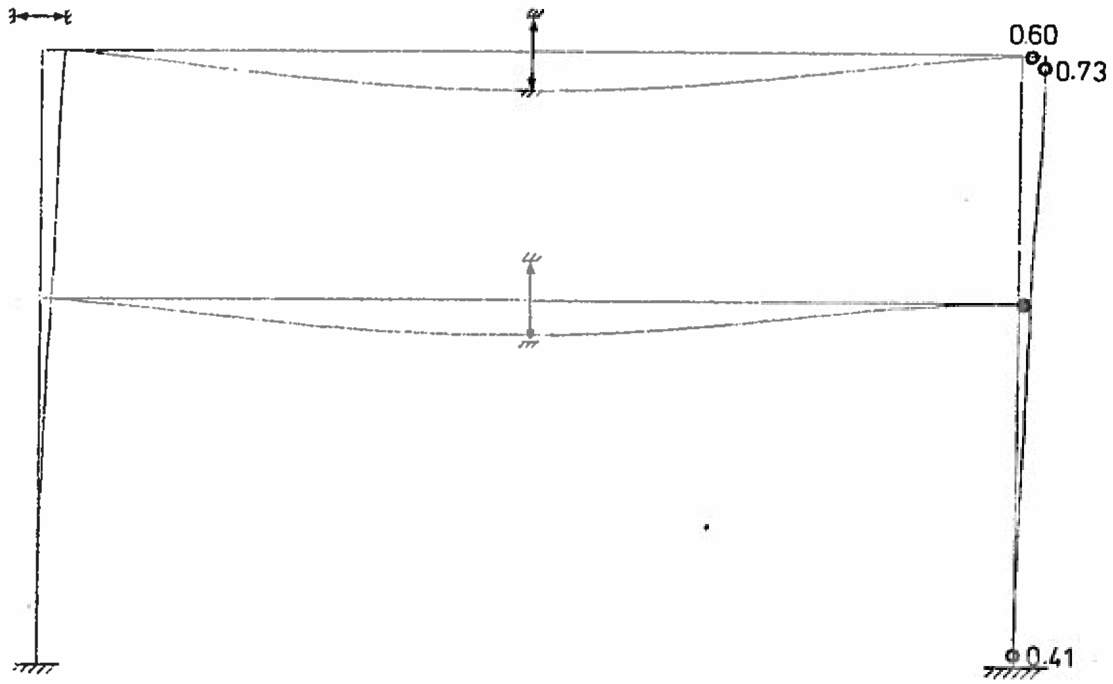


Fig. 6. Frame OH-2E1FU, most critical values of φ_s within each group. Displacements of loadcase no. 1, x20.

5.2 Example 2, 10 storey 3 bay.

This example is first run as a sway frame problem named OH-10E3F. Later it is redefined as a braced frame problem named OH-10E3FG.

Three loadcases are treated including two lateral loadcases.

The lateral displacement of the top right corner is restricted to 1/600 of the total height.

Providing symmetry and a suitable reduction in the number of design variables the structure is divided into 6 beam groups and 6 column groups plus in OH-10E3FG additionally 3 bracing groups.

The initial common design of all columns is taken as 200 cm^2 and the initial common design of all beams is taken as 100 cm^2.

5.2.1 Frame OH-10E3F, the sway frame.

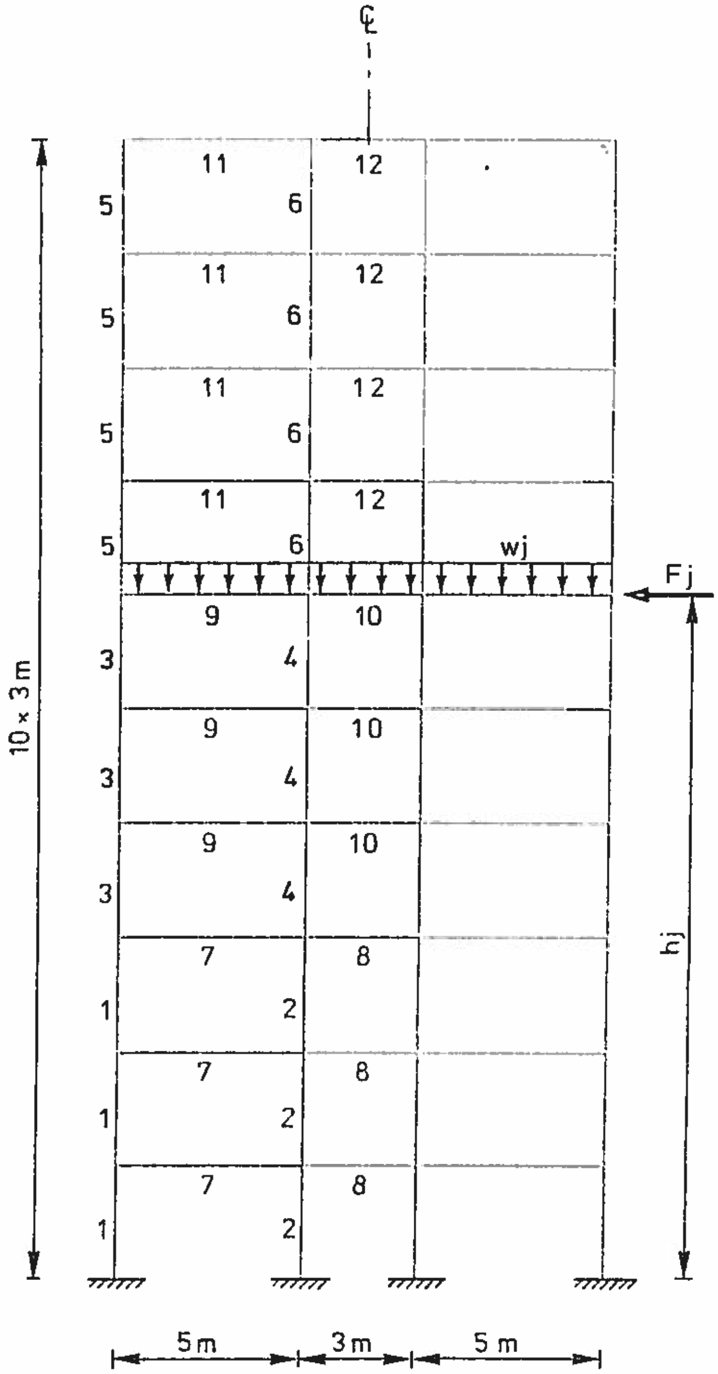
The graphic design report fig. 8 shows first 2 convergent fully stressed iterations with MFI ≃ 1.50 corresponding to the displacement constraint at iteration no. 2. The common scaling step brings MFI approximately back to 1. The following 5 minimum volume redesigns shift the distribution of stiffness in favor of a strong central core shown by the increase of all even numbered groups and the decrease of all uneven numbered groups.
At the end of stage 1 only 3 stress constraints and 1 displacement constraint are active which means that stage 2 will have only 4 active groups. Groups 1, 2 and 3 are fully stressed and therefore should be active. Group no. 6 is chosen as the fourth active group. While stage 1 has hit a very weak optimum, stage 2 shows rapid convergence to a strong optimum.

The design is completely governed by constraints from loadcase no. 2 and the stiffness distribution must be viewed as a compromise between the displacement constraint demanding a stiff structure and the equivalent earthquake forces favoring a flexible structure.

The final discrete design used as the initial design in stage 2 shows a total steel volume of 3.00 m^3 and a fundamental period of 1.58 sec. The optimal discrete design is shown below.

Group no.	Optimal design area (cm^2)	Discrete profile	Profile area (cm^2)
1	119.75	HEB 280	131
2	231.83	HEB 500	239
3	106.42	HEB 240	106
4	181.00	HEB 360	181
5	91.00	HEB 220	91
6	121.42	HEB 260	118
7	84.50	IPE 400	84.5
8	156.00	IPE 600	156
9	98.80	IPE 450	98.8
10	134.00	IPE 550	134
11	98.80	IPE 450	98.8
12	72.70	IPE 360	72.7

Table 8. Discretized optimal design of frame OH-10E3F.



Beams: IPE profiles

Columns: HEB profiles

Loadcases, common.

Distributed floor loads:

w_j = 45 kN/m (j = 1, 10)

Gravity load: γ = 78 kN/m³

Lateral deflection of top right corner $\leq$ 1/600 of height.

Loadcase no. 1.

$\sigma^N, \sigma^B = 1.7 \cdot 10^5$ kN/m²

Loadcase no. 2.

Additional earthquake loads:

$$F_j = C \cdot \frac{0.05}{\sqrt[3]{T}} g \cdot \frac{h_j WM_j}{\Sigma h_i WM_i} \Sigma WM_k \; (j=1,10)$$

WM_j is total mass at floor j

g is accel. of gravity

T is first period of frame

C = 1.00

$\sigma^N, \sigma^B = 1.9 \cdot 10^5$ kN/m²

Loadcase no. 3.

Additional wind loads:

$$F_j = C \cdot \frac{A_j}{16} (12(\log_{10} h_j + 1.3))^2 g 10^{-3} \text{kN} \quad (j=1,10)$$

C is shapefactor = 1.8

A_j is exposed area at floor j (6 m between frames)

$\sigma^N, \sigma^B = 1.9 \cdot 10^5$ kN/m²

Fig. 7 Frame OH-10E3F, loadings, group numbers and geometry.

OH-10E3F NBEL=3 MAX J/L=1/600 NCDIS=1 C=1.00 UNIT=KN,M
ISTAD= 0 NAGRUP=12 NEDI= 3 [illegible] 0.70 REDUKF= 0.70 [illegible] 0.20

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2

	0	1	2	3	4	5	6	7	8
1	200.00	-172.73	-139.57	-1[illegible].77	-157.32	-135.[illegible]	124.89	121.1[illegible]	130.17
2	200.00	-192.91	-191.65	-256.25	-189.03	205.21	120.59	22[illegible].4[illegible]	230.26
3	200.00	-138.71	-125.13	-154.[illegible]	-123.41	114.[illegible]	110.91	10[illegible].7[illegible]	1[illegible]5.[illegible]
4	200.00	-148.35	-141.57	-174.4[illegible]	-139.52	159.13	174.72	1[illegible].71	1[illegible].47
5	200.00	-113.93	-94.2[illegible]	-117.11	-9[illegible].6[illegible]	106.[illegible]	-93.32	-[illegible].7[illegible]	-[illegible]4.03
6	200.00	-117.25	-101.37	-125.6[illegible]	-130.43	114.56	125.77	-134.[illegible]0	-127.94
7	100.00	-84.1[illegible]	-81.11	-100.[illegible]	120.00	-103.2[illegible]	-9[illegible].09	-86.70	-[illegible]2.54
8	100.00	-74.37	-68.96	-84.90	101.86	116.14	127.32	13[illegible].27	142.32
9	100.00	-83.72	-80.74	-99.5[illegible]	119.46	-102.71	107.05	-[illegible]	-94.90
10	100.00	-69.37	-62.27	-76.77	92.12	105.02	115.31	123.2[illegible]	129.14
11	100.00	-77.12	-72.25	-89.6[illegible]	9[illegible].35	103.7[illegible]	-90.25	[illegible]5.[illegible]1	-[illegible]2.76
12	100.00	-56.12	-47.66	-58.7[illegible]	70.51	80.[illegible]	-72.50	-[illegible]7.53	-70.77
S	3.7000	2.7353	2.5344	[illegible].124[illegible]	[illegible].9093	2.9[illegible]	2.9364	2.9707	2.9119
MFI	0.9494	1.3243	1.5200	1.0211	1.0951	1.0234	1.0046	1.0043	1.0038
T	1.4831	1.9530	2.1004	1.6445	1.695[illegible]	1.5365	1.6209	1.61[illegible]	1.6177

Table 9. Design report, stage 1.

```
OH-10E3F NBEL=3 MAX U/L=1/600 NCDIS=1  C=1.00
 ISTAE= 5 NAGRUP= 4 VFSI= 0 RELAXF= 0.70 REDUKF= 0.70 FFM= 0.20          UNITS=KN,M
```

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2

	0	1	2	3
1	131.00	118.82	119.74	119.75
2	239.00	231.29	231.56	231.83
3	106.00	106.53	106.40	106.42
4	181.00	-181.00	-181.00	-181.00
5	91.00	-91.00	-91.00	-91.00
6	131.00	119.79	121.45	121.42
7	84.50	-84.50	-84.50	-84.50
8	156.00	-156.00	-156.00	-156.00
9	98.80	-93.80	-98.80	-98.80
10	134.00	-134.00	-134.00	-134.00
11	98.80	-98.80	-98.80	-98.80
12	72.70	-72.70	-72.70	-72.70
S	3.0087	2.9470	2.9534	2.9534
MFI	1.0067	1.0081	1.0001	1.0000
T	1.5820	1.6121	1.6086	1.6087

Table 10. Design report, stage 2.

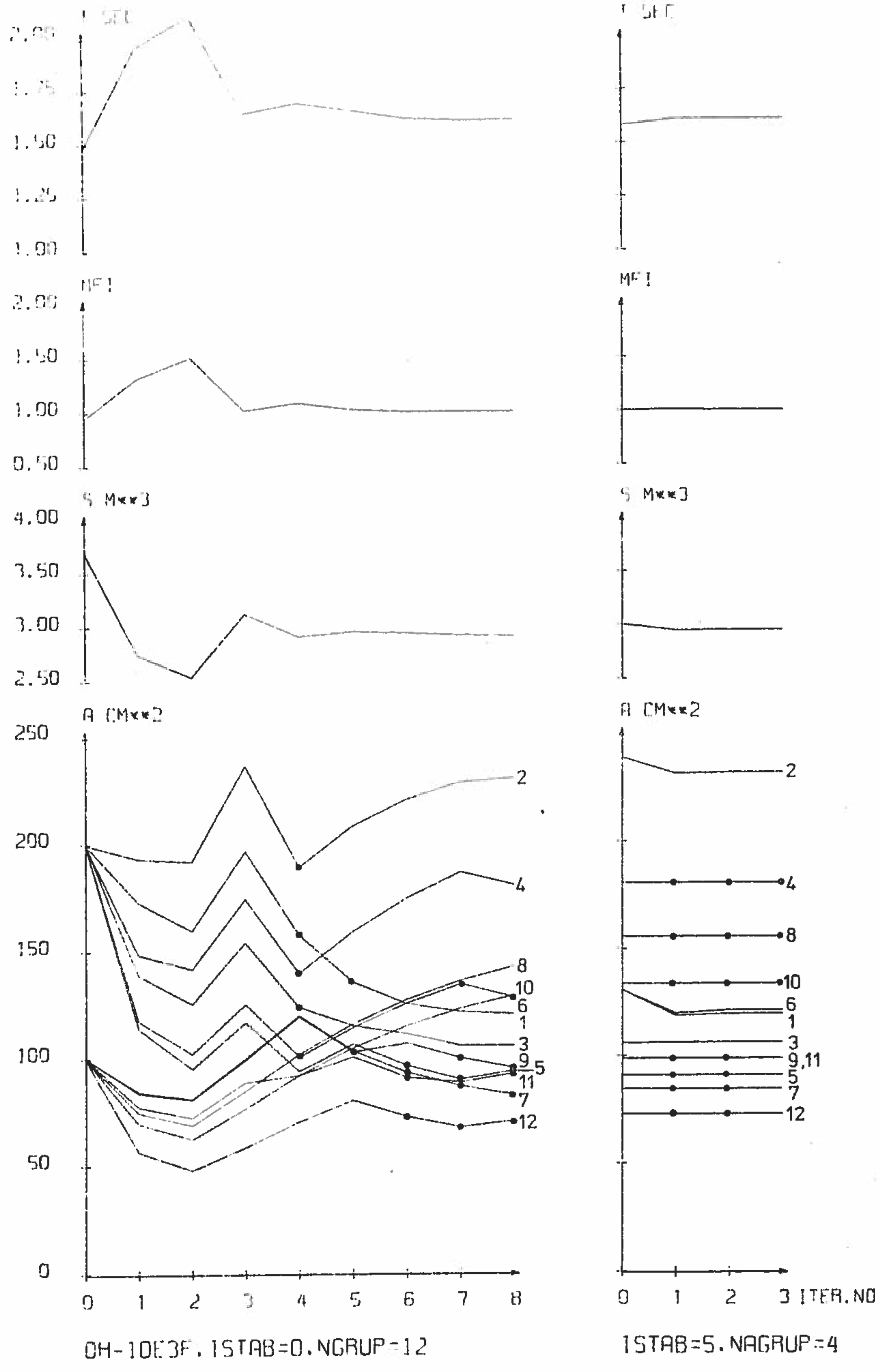


Fig. 8 Graphic design report, stage 1 and stage 2.

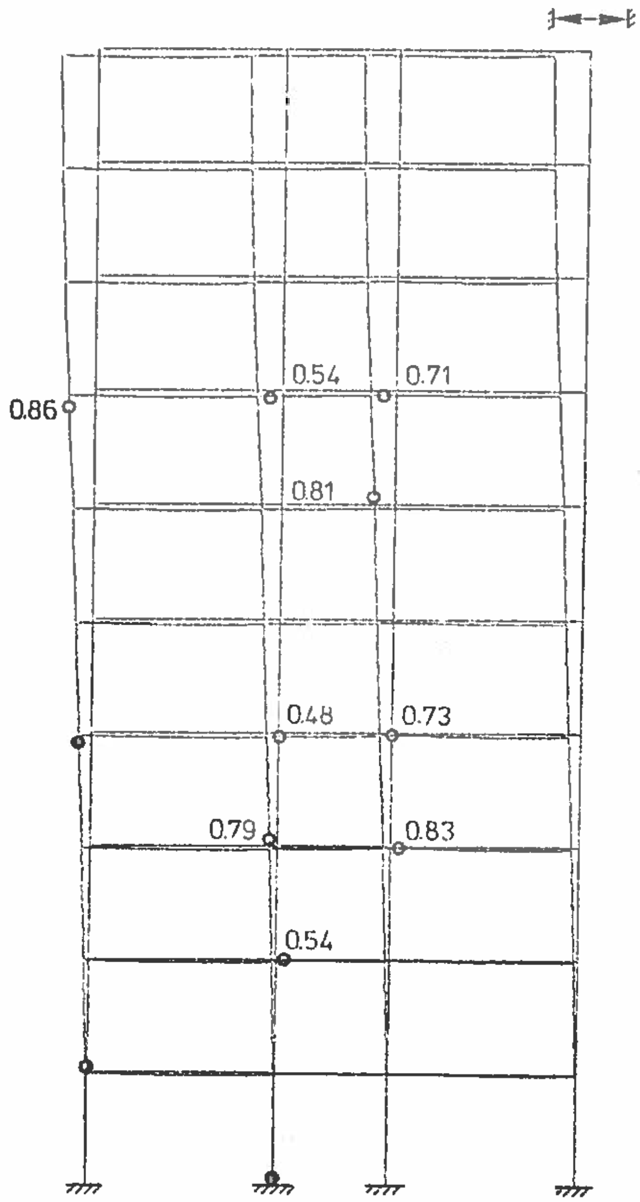


Fig. 9 Frame OH-10E3F. Most critical values of φ_s within each group. Displacements of loadcase no. 2, x 20.

5.2.2 Frame OH-10E3FG, the braced frame.

In order to appraise a braced frame load carrying system as opposed to a sway frame load carrying system diagonal bracing was inserted between the central columns. By the AUTOVNF = 1 option in the computer program automatic prestress of the bracing at half the allowable tension under the vertical loadcase, was accomplished
Because of the presumably smaller energy absorption capacity of a braced frame as opposed to a sway frame a 33% higher value of C in the equivalent earthquake loads is used. Braces are chosen from a solid circular geometrically similar series with initial values of 10, 20 and 30 cm^2. Aside from bracing and the higher value of C frame OH-10E3FG is identical to frame OH-10E3F.

The graphic design report of the braced frame, fig. 11, shows a much more smooth iteration path than that of the sway frame. The variables especially partaking in displacement control are seen to be 2 of the 3 bracing groups, the 2 upper outer beam groups and the upper outer column group. All the other groups are nicely fully stressed.
An examination of the set of active constraints of the end of stage 1 reveals 10 active stress constraints and 1 active displacement constraint. With 15 groups this indicates a weak optimum. Stage 2 therefore has only 11 active groups. Out of the 5 displacement controlling groups 4 are deactivated and 1, namely group 5, the upper outer column, is kept active. This gives a rapid convergence of stage 2 to a strong optimum.

The optimum design is completely controlled by constraints from loadcase no. 2 except group 10 which is stress contrained in load-case no. 1.

The final discrete design used as the initial design in stage 2 shows a total steel volume of 2.88 m^3 and a fundamental period of T = 1.29 sec.
In comparison with the sway frame we now have a frame using 4% less material even with a 33% higher lateral load coefficient and even with a smaller fundamental period which induces still larger lateral loads. Thus bracing introduces lateral stiffness which more than offsets the induced larger lateral loads.

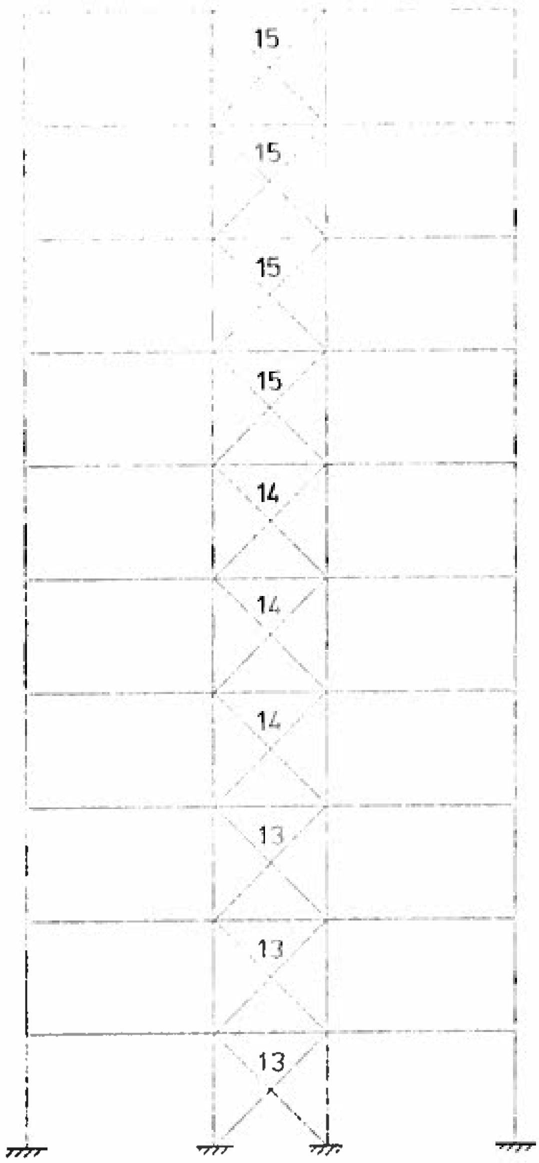
When comparing the two kinds of frames, however, one should also consider the added cost of erection which may well exceed the saved cost of material.

In neither 10 storey frame did nonlinear geometry in analysis show noticeable effects.

The optimal discrete design is shown below.

Group no.	Optimal design area (cm^2)	Discrete profile	Profile area (cm^2)
1	127.14	HEB 280	131
2	234.91	HEB 500	239
3	122.40	HEB 260	118
4	133.65	HEB 280	131
5	89.28	HEB 240	106
6	83.44	HEB 220	91
7	70.58	IPE 360	72.7
8	58.10	IPE 330	62.6
9	116.00	IPE 500	116
10	43.02	IPE 270	45.9
11	98.80	IPE 450	98.8
12	44.19	IPE 270	45.9
13	33.00	GS 33	33
14	16.05	GS 20	20
15	15.00	GS 15	15

Table 11. Discretized optimal design of frame OH-10E3FG



Loadcases.

Same as for OH-10E3F except the earthquake loads, where C = 1.00 is increased to C = 1.33 because of the difference in load carrying systems.

Bracing: Geometrically similar series (prestressed)

Fig. 10. OH-10E3FG, additional group numbers and specifications.

```
OH-10E3FG NBEL=3 MAX U/L=1/600 NCDIS=1  C=1.00
 ISTAB= 0 NAGRUP=15 NFSI= 3 RELAXF= 0.70 REDUKF= 0.70 FFM= 0.20      UNITS=KN,M

DESIGN REPORT
DOWN:GROUPNO.,AREA IN CM**2,RIGHT: ITERATIONNO.
```

	0	1	2	3	4	5	6	7	8
1	200.00	-162.63	-140.99	-143.85	122.99	127.18	126.39	126.42	126.40
2	200.00	-210.45	-222.31	-234.70	238.80	234.69	234.61	235.21	235.18
3	200.00	-145.10	-125.38	-132.37	115.63	117.59	112.25	119.73	119.29
4	200.00	-152.76	-138.75	-146.46	127.71	131.60	132.13	133.31	132.66
5	200.00	-121.47	-101.41	-107.05	90.50	103.52	-93.47	83.74	93.00
6	200.00	-119.23	-96.49	-101.37	84.51	79.92	83.07	81.92	82.24
7	100.00	-83.97	-73.35	-77.44	69.82	70.30	60.35	70.14	70.05
8	100.00	-68.95	-58.50	-61.76	54.77	56.60	57.99	56.99	57.65
9	100.00	-88.45	-79.89	-84.34	101.21	115.33	112.69	113.96	-114.20
10	100.00	-63.81	-50.06	-52.85	44.61	45.79	45.13	45.04	45.16
11	100.00	-86.97	-77.17	-81.47	94.85	-31.59	89.57	93.87	89.13
12	100.00	-52.99	-42.81	-45.19	41.83	43.55	42.44	42.77	43.48
13	30.00	-28.57	-27.46	-28.99	26.84	30.59	33.59	-31.29	32.79
14	20.00	-21.06	-21.57	-22.77	20.71	20.95	20.01	19.89	20.02
15	10.00	-10.64	-11.37	-12.00	12.65	14.42	-13.01	13.90	14.57
S	3.8612	2.9963	2.6852	2.8349	2.7066	2.7520	2.7562	2.7624	2.7627
MFI	1.0919	1.0970	1.1146	1.0430	1.0360	1.0129	1.0050	1.0015	1.0008
T	1.1656	1.3230	1.4152	1.3542	1.3599	1.3308	1.3272	1.3230	1.3240

Table 12. Design report, stage 1.

```
CH-10E3FG NBEL=3 MAX U/L=1/600 NCDIS=1   C=1.00                                   UNITS=KN,M
 ISTAB= 5 NAGRUP=11 NFSI= 0 RELAXF= 0.70 REDUKF= C.70 FFM= 0.20

DESIGN REPORT
DOWN:GROUPNO.,AREA IN CM**2,RIGHT: ITERATIONNO.
```

	0	1	2	3
1	131.00	127.09	127.14	127.14
2	239.00	234.92	234.69	234.71
3	118.00	122.26	122.41	122.40
4	131.00	133.66	133.63	133.65
5	106.00	87.98	89.27	89.28
6	91.00	83.35	83.44	83.44
7	72.70	70.50	70.58	70.58
8	62.60	57.73	58.09	58.10
9	116.00	-116.00	-116.00	-116.00
10	45.90	42.96	43.01	43.02
11	98.80	-98.80	-98.80	-98.80
12	45.90	44.08	44.19	44.19
13	33.00	-33.00	-33.00	-33.00
14	20.00	-16.00	16.02	16.05
15	15.00	-15.00	-15.00	-15.00
S	2.8869	2.7974	2.8017	2.8013
MFI	1.0229	1.0109	1.0002	1.0000
T	1.2994	1.3327	1.3311	1.3310

Table 13. Design report, stage 2.

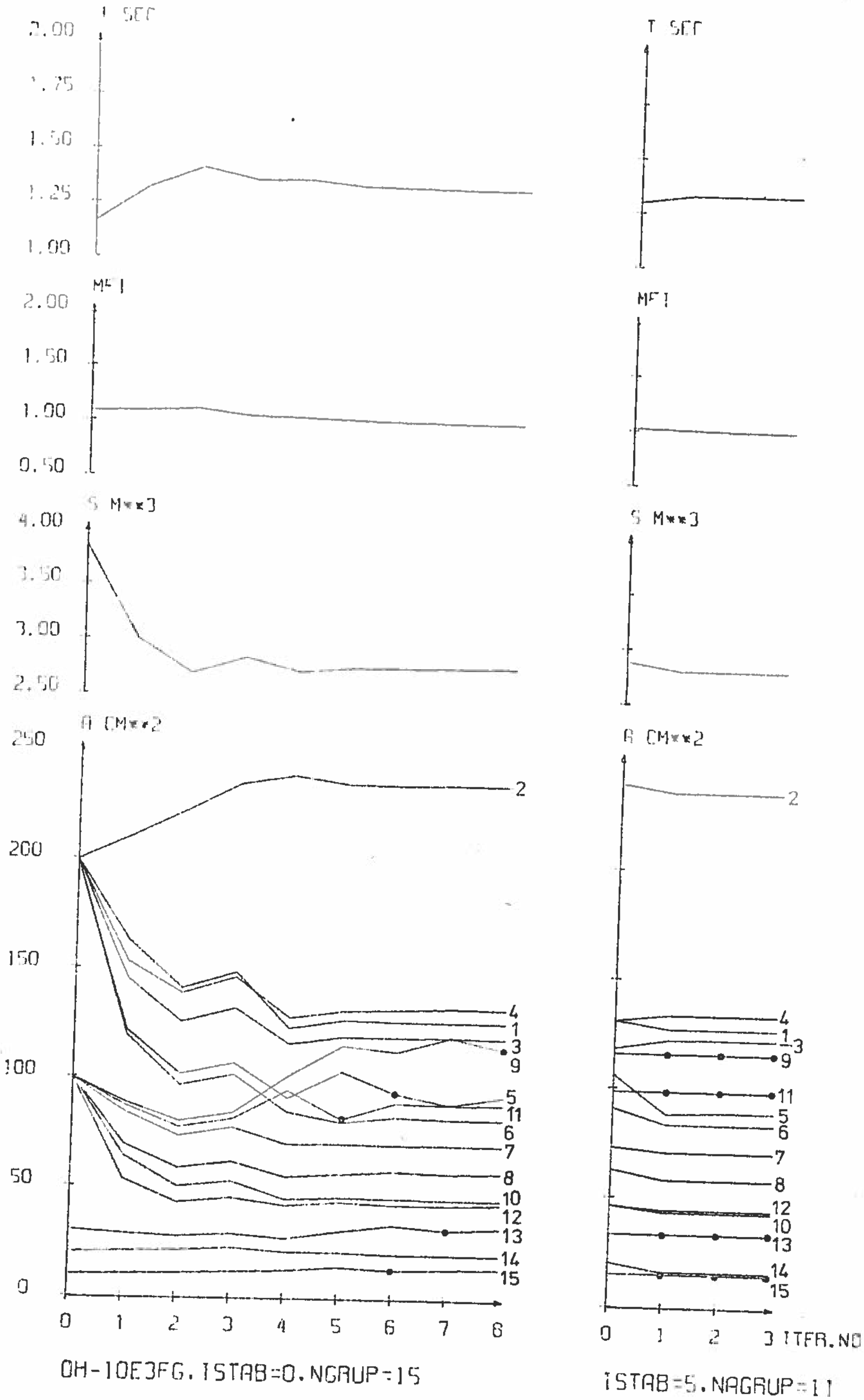


Fig. 11. Graphic design report, stage 1 and stage 2.

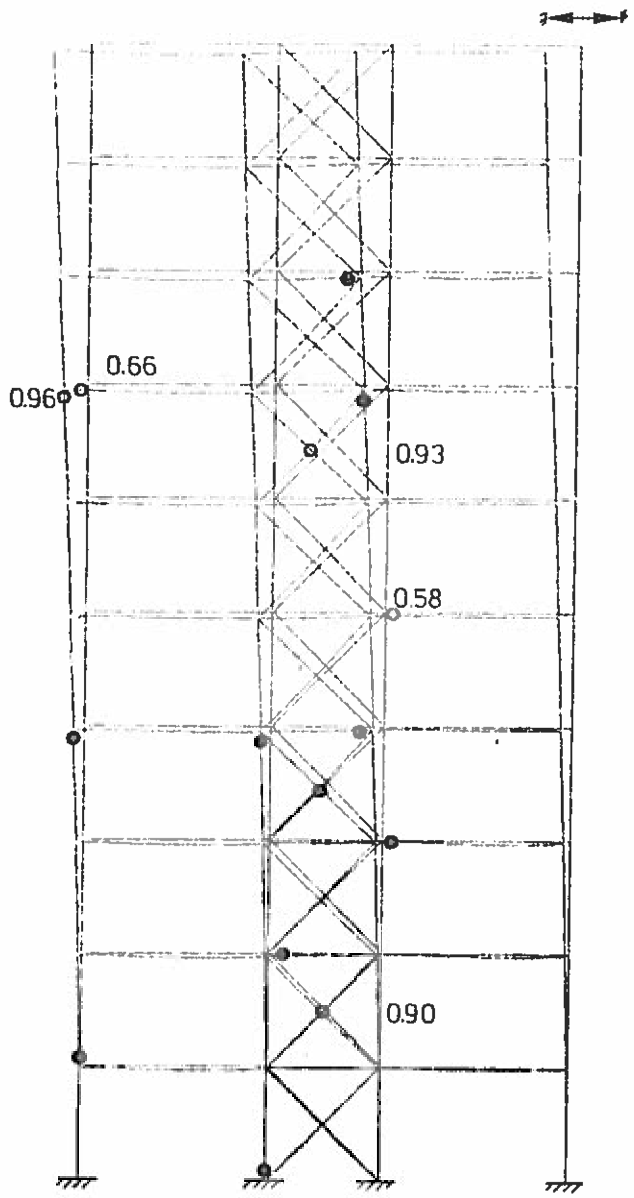


Fig. 12. Frame OH-10E3FG, most critical values of φ_s within each group. Displacements of loadcase no. 2, x20.

5.3 Example 3, 30 storey 2 bay.

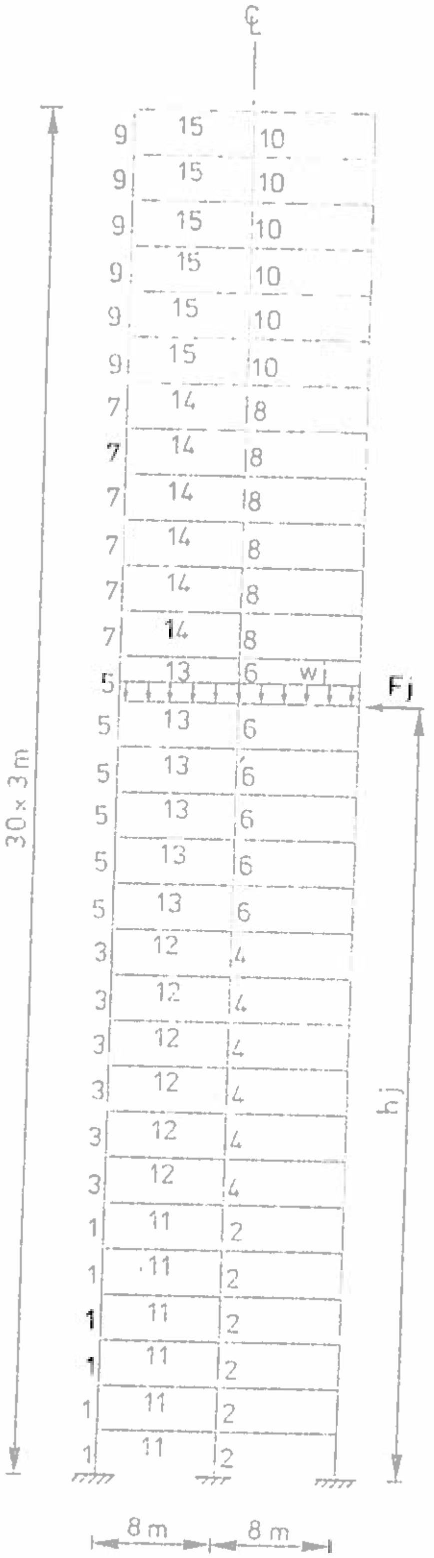
This frame named OH-30E2F is a high rise building frame with 150 elements. Symmetry plus suitable grouping, however, reduces the number of design groups to 15. Two loadcases are treated. No. 1 with vertical loads only and no. 2 with vertical loads plus lateral earthquake loads as defined on fig. 13. The lateral displacement of the top is limited to 1/300 of the total height.
The initial design of the lower column groups is 300 cm^2 and of the upper columns 200 cm^2. The beams are all initially at their maximal value 156 cm^2.
After the initial 2 convergent fully stressed iterations the graphic design report, fig. 14, shows a rather chaotic oscillating design process bearing the mark of a very weak optimum. Accordingly, at the end of stage 1 only the 3 lower central column groups are fully stressed in loadcase no. 1 while the remaining 12 groups are designed on the displacement constraint of loadcase no. 2. Because of the decreasing movelimit the oscillations at the end of stage 1 are starting to fade out and intermediate discrete values may be chosen for all design variables as an initial design for stage 2. The second stage has only 4 active variables, namely the 3 fully stressed plus the upper beam no. 15 which is chosen as the displacement controller.
Stage 2 now shows the customary rapid convergence to a strong optimum. The total steel volume exhibits a jump from the final value of stage 1 to the final value of stage 2 which in this high rise frame is attributable to the nonlinear geometric analysis used in stage 2. Stage 2 run with linear geometric analysis yields a steel volume of 12.06 m^3 while nonlinear analysis requires 12.32 m^3, an increase of 2%. The values of the upper beams no. 15 in the 2 cases are 80.05 cm^2 and 107.33 cm^2 respectively and the greatest changes in φ_s values (stress levels) occur in the beams.

The overturning moment at any floor level is restricted partly by moment action in the columns, partly by normal force action in the outer columns. The distribution of moment stiffness in the frame determines how much of the turning moment will be transmitted into normal forces in the outer columns. An investigation of the stress distribution based on the optimal stiffness distribution shows that up to 95% of the overturning moment is taken by normal forces in the columns. Hence it may be concluded that in high rise sway frames efficient lateral displacement control requires a stiffness distribution that transmits the overturning moment into normal forces in the outer columns.

The final discrete design using nonlinear geometric analysis has a steel volume of 12.28 m^3 and a fundamental period of 3.71 sec. The individual design values are shown below.

Group no.	Optimal design area (cm^2)	Discrete profile	Profile area (cm^2)
1	306.00	HEB 700	306
2	385.78	HEB 1000	400
3	239.00	HEB 500	239
4	303.96	HEB 700	306
5	181.00	HEB 360	181
6	230.24	HEB 500	239
7	131.00	HEB 280	131
8	198.00	HEB 400	198
9	91.00	HEB 220	91
10	161.00	HEB 320	161
11	156.00	IPE 600	156
12	156.00	IPE 600	156
13	134.00	IPE 550	134
14	134.00	IPE 550	134
15	107.33	IPE 450	98.8

Table 14. Discretized optimal design of frame OH-30E2F.



Loadcases, common.

Specific gravity: $\gamma = 78\ \mathrm{kN/m^3}$

Loadcase no. 1.

Floor loads: $w_j = 25\ \mathrm{kN/m}$ (j=1,30)

$\sigma^N_{IPE} = 1.25 \cdot 10^5\ \mathrm{kN/m^2}$ $\sigma^N_{HEB} = 1.89 \cdot 10^5\ \mathrm{kN/m^2}$

$\sigma^B_{IPE} = 1.57 \cdot 10^5\ \mathrm{kN/m^2}$ $\sigma^B_{HEB} = 2.36 \cdot 10^5\ \mathrm{kN/m^2}$

Loadcase no. 2.

Floor loads: $w_j = 18.75\ \mathrm{kN/m}$ (j=1,30)

Equivalent earthquake loads:

$$F_j = C\frac{0.05}{\sqrt[3]{T}} \cdot g\, \frac{h_j WM_j}{\Sigma h_i WM_i}\, \Sigma WM_k \qquad (j=1,30)$$

$C = 1.33$

$\sigma^N_{IPE} = 1.73 \cdot 10^5\ \mathrm{kN/m^2}$ $\sigma^N_{HEB} = 2.59 \cdot 10^5\ \mathrm{kN/m^2}$

$\sigma^B_{IPE} = 2.16 \cdot 10^5\ \mathrm{kN/m^2}$ $\sigma^B_{HEB} = 3.24 \cdot 10^5\ \mathrm{kN/m^2}$

Lateral displacement of top right corner $\leq$ 1/300 of height.

Beams: IPE profiles

Columns: HEB profiles

Fig. 13. Frame 30E2F, loadings, group numbers and geometry.

```
OH-30E2F  NBEL=2  MAX U/L=1/300  NCDIS=1
 ISTAB= 0 NAGRUP=15 NESI= 3 RELAXF= 0.70 REDUKF= 0.70 EEM= 0.20          UNITS:KN,M

DESIGN REPORT
DOWN:GROUPNO.,AREA IN CM**2,RIGHT: ITERATIONNO.
```

	0	1	2	3	4	5	6	7	8
1	300.00	-295.39	-262.03	-326.47	-261.17	277.74	271.74	290.38	304.33
2	300.00	-333.85	-359.57	-400.00	327.04	330.16	398.60	389.50	390.10
3	300.00	-251.30	-225.33	-261.06	-209.55	233.03	-214.75	229.48	-218.46
4	300.00	-277.60	-285.83	-330.89	310.42	235.02	314.61	306.38	307.75
5	300.00	-217.59	-195.54	-214.78	-171.82	195.93	-176.68	193.80	-179.74
6	300.00	-227.42	-218.21	-252.60	232.00	272.22	233.92	239.31	231.90
7	200.00	-156.22	-177.70	-159.40	-127.52	115.37	-131.12	140.12	-133.39
8	200.00	-151.45	-146.57	-169.78	203.73	-175.21	192.38	205.58	-195.71
9	200.00	-121.04	-96.42	-111.69	-89.35	101.36	-91.88	98.12	-93.47
10	200.00	-105.22	-83.09	-96.19	115.42	131.53	144.43	154.39	-146.98
11	156.00	-123.10	-115.35	-134.10	116.25	132.53	145.52	-135.54	142.04
12	156.00	-123.24	-116.25	-134.57	156.00	133.46	-133.42	147.92	155.02
13	156.00	-119.24	-111.86	-129.49	155.38	-133.67	146.73	-136.66	143.22
14	156.00	-108.47	-99.51	-115.19	138.23	-113.88	130.53	130.54	125.97
15	156.00	-102.85	-88.75	-102.72	-82.18	93.69	102.97	-95.81	-91.21
S	14.5090	11.2702	10.4162	12.0275	11.5782	11.7641	11.9764	11.9255	11.9469
MFI	1.1612	1.1557	1.3399	1.0506	1.0626	1.0372	1.0209	1.0089	1.0052
T	3.3425	4.1754	4.4753	3.8470	3.8888	3.8053	3.7830	3.7638	3.7000

Table 15. Design report, stage 1.

```
OH-30E2F   NBEL=2   MAX U/L=1/300   NCDIS=1
 ISTAB=  5 NAGRJP=  4 NFSI=  0 RELAXF=  0.70 REDUKF=  0.70 FFM=  0.20        UNITS:KN.M

OVERSIGTSSKEMA
ITERATIONSNR. HENAD,GRUPPENR. NEDAD,AREAL I CM**2
```

	0	1	2	3
1	306.00	-306.00	-306.00	-306.00
2	400.00	385.42	385.77	385.78
3	239.00	-239.00	-239.00	-239.00
4	306.00	303.93	303.95	303.96
5	181.00	-181.00	-181.00	-181.00
6	239.00	230.07	230.23	230.24
7	134.00	-134.00	-134.00	-134.00
8	198.00	-198.00	-198.00	-198.00
9	91.00	-91.00	-91.00	-91.00
10	161.00	-161.00	-161.00	-161.00
11	156.00	-156.00	-156.00	-156.00
12	156.00	-156.00	-156.00	-156.00
13	134.00	-134.00	-134.00	-134.00
14	134.00	-134.00	-134.00	-134.00
15	98.80	104.75	106.68	107.33
S	12.2873	12.2986	12.3198	12.3242
MFI	1.0073	1.0029	1.0006	1.0001
T	3.7158	3.7211	3.7198	3.7195

Table 16. Design report, stage 2.

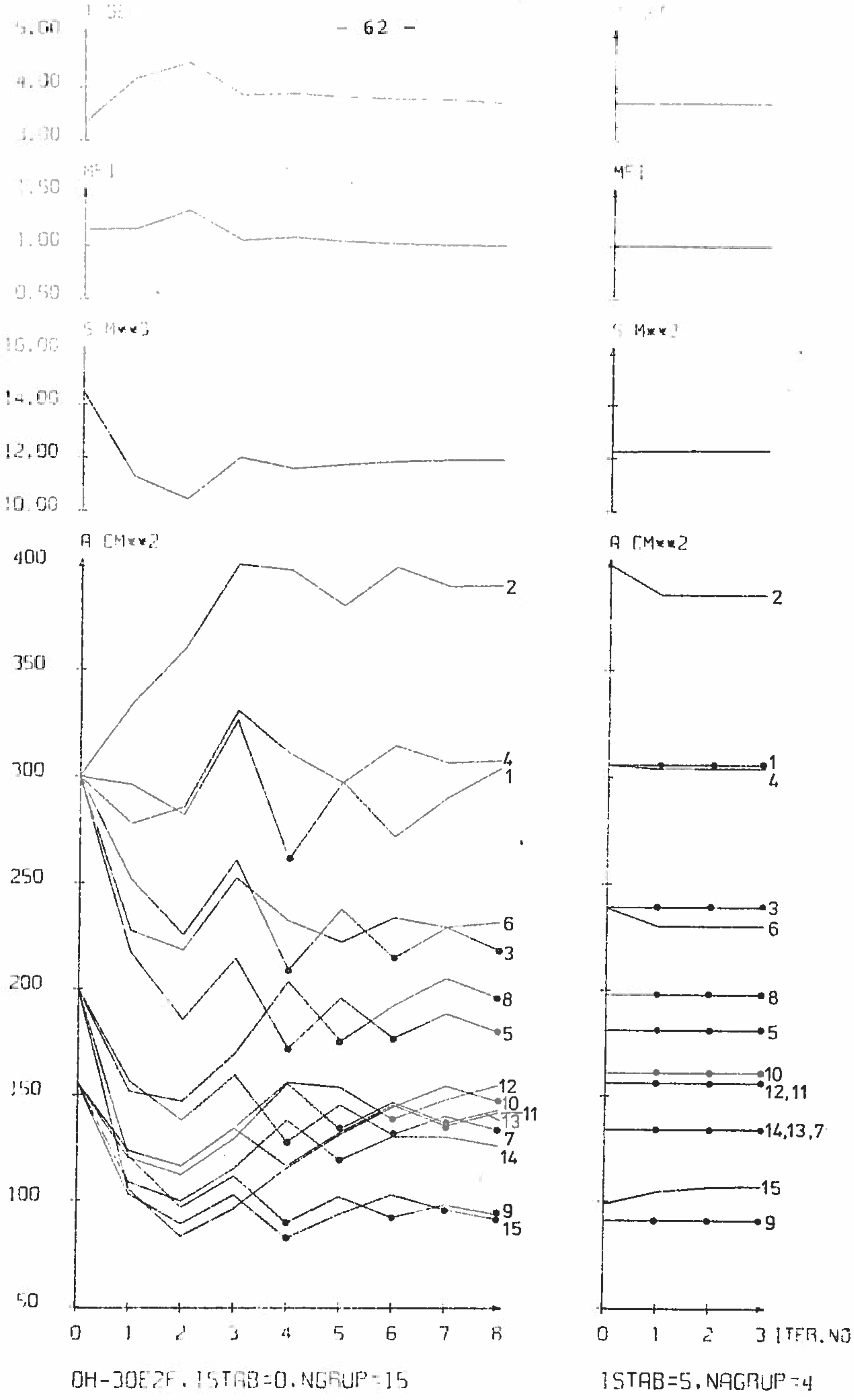


Fig. 14. Graphic design report, stage 1 and stage 2.

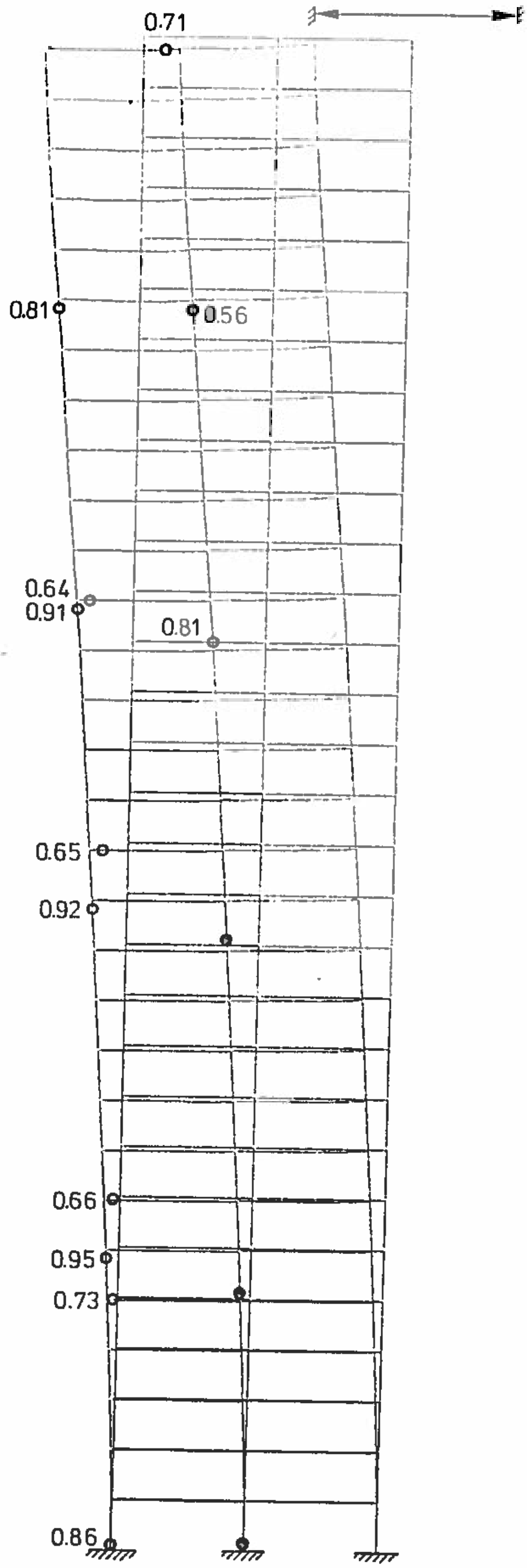


Fig. 15. Frame OH-30E2F, most critical values of φ_s within each group, displacements of loadcase no. 2, x20.

6 Conclusions.

The recommended design strategy of chapter 4 which calls for two fully stressed iterations, one common scaling for maximum constraint value and finally a number of minimum volume redesigns using sequential linear programming, has been shown through the presented design examples to produce feasible minimum volume designs with very little human intervention once the problem has been properly stated.

Structures with large numbers of elements may be designed by the method, but care should always be taken to restrict the number of actual design variables by suitable and realistic grouping of elements.

Using diminishing movelimits on all design variables convergence to a unique design is always obtained, albeit perhaps slowly to a weak optimum. In case of a weak optimum restating the problem with the correct set of active design variables guarantees rapid convergence to a strong optimum, i.e. to a unique design without the help of movelimits.

Design dependent loads like selfweight, equivalent earthquake loading and extra loading due to nonlinear geometric analysis have been considered without posing any threat to the stability of the convergence of the designs.

The recommended design strategy is perfectly general and may easily be extended to include design of any structure amenable to finite element analysis and having a finite number of discrete structural constraints.

Although minimum volume mathematically is the design objective, minimum volume is not the main virtue of the presented method, but rather the ways and means of automatically obtaining a unique, feasible design which also happens to use less material than any other possible feasible design with the same geometric layout.

In other words, rational, automatic design is deemed more virtuous than optimality of design.

7 References.

[1] Gellatly, R.A. & Berke, L.: "Optimal Structural Design" Wright Patterson AFB, AFFDL-TR-70-165, 1971

[2] Fox, R.L.: "Optimization Methods for Engineering Design", Addison-Wesley P.C., 1971.

[3] Holst, Ole: "Linear Programmering, med EDB program og eksempler",Structural Research Laboratory, F43, Technical Univ. of Denmark, 1972.

[4] Razani, Reza: "Behavior of Fully Stressed Design of Structures and Its Relationship to Minimum-Weight Design", AIAA Journal, Vol. 3, No. 12, 1965. pp. 2262-2268.

[5] Kicher, T.P.: "Optimum Design - Minimum Weight versus Fully Stressed", Journal of the SD, ASCE, Vol. 92, No. ST6, Dec. 1966. pp. 265-279.

[6] Holst, Ole: "Beregning af plane rammekonstruktioner med geometrisk ikkelinearitet", Structural Research Laboratories, R41, Technical University of Denmark, 1973.

[7] Seismology Committee of Structural Engineers Association of California: "Recommended Lateral Force Requirements", Dec. 1959.

[8] Pedersen, Pauli: "Optimal layout af gitterkonstruktioner", The Danish Center for Applied Mathematics and Mechanics, DCAMM nr. S1. Technical University of Denmark, Sept. 1970.

[9] Solnes, Julius & Holst, Ole: "Weight optimization of Framed Structures under Earthquake Loads", Structural Research Laboratory, R33, Technical Univ. of Denmark, 1972.

[10] Niordson, F.I. & Pedersen, Pauli: "A Review of Optimal Structural Design". Department of Solid Mechanics, Technical University of Denmark, 1972.

[11] Romstad, K.M. & Wang, C-K.: "Optimum Design of Framed Structures", Journal of the SD, ASCE, Vol. 94, No. ST12, Dec. 1968, pp. 2817-2845.

[12] Livesley, R.K.: "Linear Programming in Structural Analysis and Design", International Symposium on Computers in Optimisation of Structural Design, University of Wales, Swansea, Jan. 1972.

Appendix 1. Description of the frame analysis.

Table of contents. Page

Notation

A,AREA	Element area.
A_1, A_2, A_3	Stiffness constants.
B_1	Stiffness constant.
C	Structure factor in earthquake loading.
$C_0 - C_5$	Stiffness constants.
E,EMOD	Young's modulus.
F,BNO	Nodal force or load, global datum.
F^o	Initial load, global datum.
I,IMOM	Moment of inertia.
IA,IB	Global node numbers of element.
IBEL	Load case number.
IDIS	Displacement number.
IELM	Element number.
IENS	Geometrically alike group number.
IGRUP	Design group number.
IKE	End fixity type number.
INO	Node number.
IPROF	Profile type number.
ISNIT	Stress check point number in element.
$\underset{\sim}{K}_{AA}$	Submatrix of element stiffness, local datum.
$\underset{\sim}{K}_{IAIA}$	Submatrix of element stiffness, global datum.
LENS	List of IENS'es
LGRUP	List of IGRUP's.
LKE	List of IKE's.
LNSNIT	List of NSNIT's.
LPROF	List of IPROF's.
M	Moment in element.
N	Normal force in element.
NBEL	Number of loadcases.
NELM	Number of elements.
NENS	Number of geometrically alike elements.
NNO	Number of nodes.
NSNIT	Number of stress check points in an element.
S	Nodal force or load, local datum.
S^o	Initial S , local datum.
T	Natural period.
TOPO	Element connectivity.
UST	Maximum allowable displacement.
W,WMOM	Modulus of section.
WM	Lumped mass.

d	Stiffness constant.
f	Functional relation.
g	" "
kl	Stiffness parameter.
k_1-k_5	Stiffness parameters.
l,SL	Geometric length of element.
p	Total distributed load on element.
r	Slenderness ratio.
r,x	Displacements, global datum.
u_2	Transversal displacement, local datum.
v	Displacements, local datum.
v_{Nf},VNF	Lack of fit.
w,PJ	Constant distributed load on elements.
x_1,X1	Node coordinate.
x_2,X2	Node coordinate.
δ_G	Geometrical shortening.
δ_{Gf}	δ_G due to initial curvature $\varkappa_f$.
γ,GAMMA	Specific gravity.
$\varkappa_f$	Initial constant curvature of element.
λ	Slenderness ratio.
λ_1,λ_2,A	Direction cosines.
μ	Bending stiffness.
ξ	Dimensionless distance to stress check point.

Special notation.

(j=1,M)	j = 1,2,3,4 ... M
~	vector or matrix.

A1-1 The nodes.

The frame consists of NNO nodes connected by NELM straight beam elements.

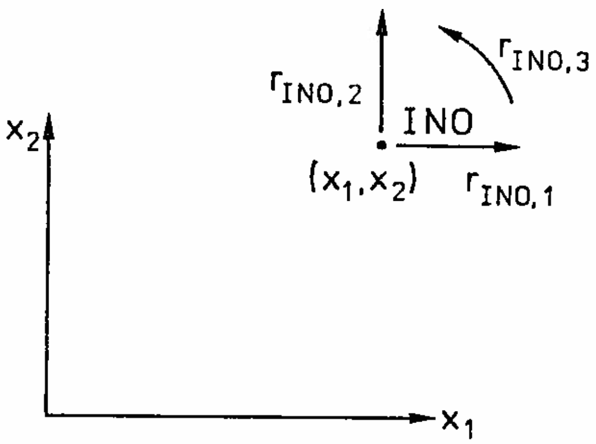


Fig. A1-1 Global frame of reference and nodal displacements.

Each node is given a global node number INO $\in$ (1,NNO) and its position is specified by the cartesian coordinates x_1, x_2 where

$$x_1 = X1(INO) \qquad (A1-1)$$
$$x_2 = X2(INO)$$

At each node 3 degrees of freedom or displacements exists. Directions and numbering is as shown on fig. A1-1.

The displacements at node INO under loadcase IBEL are defined by

$$r_{INO,i} = X(IBEL,IDIS) \qquad (A1-2)$$

where

$$IDIS = 3 \times INO - 3 + i \qquad (i=1,3) \qquad (A1-3)$$

The total number of displacements is seen to equal 3 x NNO.

Supports and internal hinges give rise to superfluous degrees of freedom at specific nodes.
Superfluous displacements are specified by assigning UST(IBEL,IDIS)=0.

The corresponding displacements are set to zero.

Support directions must be specified. Internal hinges (only hinged elements joining at the node) are recognized automatically in procedure LÆS and the appropriate elements in the array UST are set to zero.

A1-2. The elements.

In order to specify an element, three categories of information is needed, namely topology, geometry and stiffness.

A1-2.1 The topological information is by definition numbering information unaltered by variations in geometry. It is also unaltered by variations in stiffness.

The foremost topological information is the connectivity of the elements.

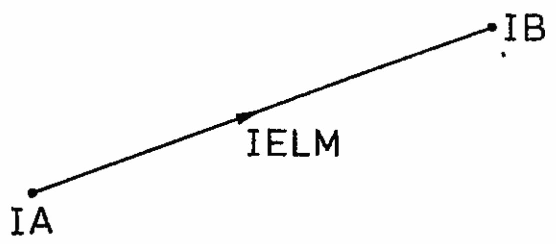


Fig. A1-2

The connectivity array TOPO gives the global node numbers of elements, defined by

$$\begin{aligned} IA &= TOPO\ (IELM,1) \\ IB &= TOPO\ (IELM,2) \end{aligned} \qquad (A1-4)$$

Many other element specifications have topological nature. The following specifications are included in the topological information.

End fixity.

There are four optional types of end fixity, designated by parameter IKE = LKE(IELM)

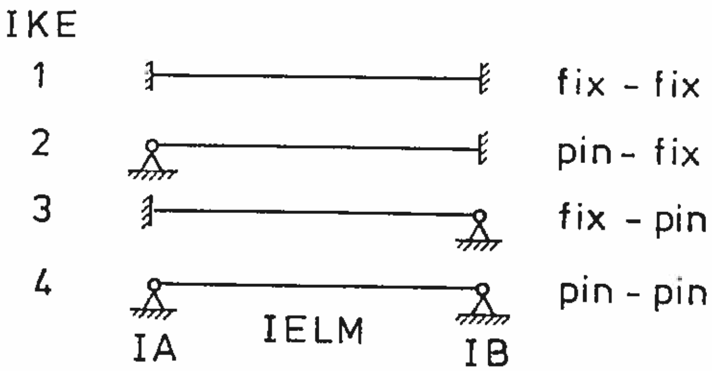


Fig. A1-3. End fixities.

At present only IKE = 1 and IKE = 4 are fully implemented.

Group design.

For reasons like symmetry and continuity of profile sections through several elements one wishes to be able to force groups of elements to have the same area in the solution. Hence elements are grouped into NGRUP different groups. The connection is given by IGRUP = LGRUP(IELM). The common area of elements with identical group numbers then serves as one design variable. The concept of group design is also important in reducing the total number of design variables.

Profile types.

One of four different profile types may be specified for each group of variables by setting IPROF = LPROF(IGRUP) = 1, 2, 3 or 4. The first three profile types are taken from the EURONORM profile series and the fourth is a theoretical, geometrically similar series (GS). Cfr. appendix 2.

IPROF	Profile
1	IPE
2	HEA
3	HEB
4	GS

Fig. A1-4. Profile types.

Stress check points.

Stress constraints are checked at NSNIT equidistantly spaced stations along each element. The stations are defined by the dimensionless distance, ξ, from the start of the element, where

$$\xi = (\text{ISNIT}-1)/(\text{NSNIT}-1) \qquad (\text{ISNIT}=1,\text{NSNIT}) \qquad (A1-5)$$

and

$$\text{NSNIT} = \text{LNSNIT(IELM)} \qquad (A1-6)$$

A1-2.2 Geometry of the elements.

The geometric information of the elements include the direction cosines λ_1 and λ_2, the length l and the lack of fit v_{Nf}

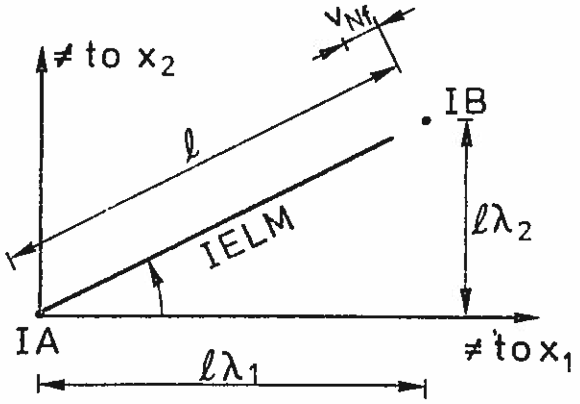


Fig. A1-5 . Element geometry.

In frames one often finds groups of elements with identical lengths and direction cosines.

Let the number of different groups of (l, λ_1, λ_2) be NENS, then define

$$l = \text{SL(IENS)} = ((\text{X1(IB)} - \text{X1(IA)})^2 + (\text{X2(IB)} - \text{X2(IA)})^2)^{\frac{1}{2}} \qquad (A1-7)$$

$$\lambda_1 = \text{A(IENS,1)} = ((\text{X1(IB)} - \text{X1(IA)})/\text{SL(IENS)} \qquad (A1-8)$$

$$\lambda_2 = \text{A(IENS,2)} = (\ \text{X2(IB)} - \text{X2(IA)})/\text{SL(IENS)} \qquad (A1-9)$$

where

$$\text{IENS} = \text{LENS(IELM)} \qquad (A1-10)$$

The lack of fit V_{Nf} = VNF(IELM) is provided to facilitate a state of geometric prestress. V_{Nf} equals the length one must elongate an unstressed element to fit it into the reference configuration.

Special geometry of pin-pin elements in compression.

When LKE(IELM) = 4 and the element is in compression the following additional geometry information is automatically assumed in the program.

According to e.g. German codes of practice, the pin-pin elements in compression are computed with a certain eccentricity of the normal force and a certain initial curvature as shown in fig. A1-6.

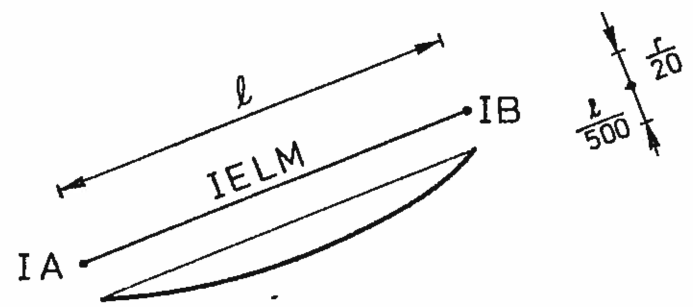


Fig. A1-6. Eccentricity and initial curvature.

The eccentricity equals r/20 where r is the radius of gyration.

The initial curvature is a constant curvature $\varkappa_f$ = -0.016/l corresponding to a maximum displacement equal to l/500.

A1-2.3 Stiffness of the elements.

Fig. A1-7. Element stiffnesses.

As explained under the element topology section, element areas of identical profile type (i.e. identical IPROF number) can be grouped to act as one design variable.
The moments of inertia I and the section moduli W are therefore grouped identically. Thus define the basic stiffness measures

$$A = AREA(IGRUP) \qquad (A1-11)$$

$$I = IMOM(IGRUP) = f\ (A,IPROF) \qquad (A1-12)$$

$$W = WMOM(IGRUP) = g\ (A,IPROF) \qquad (A1-13)$$

The functions f and g are empirical functions. Details are given in appendix 2.

Young's modulus E = EMOD, is assumed equal for all profile types and elements.

Inferred element stiffness measures include bending stiffness $\mu = EI/l$, radius of gyration $r = \sqrt{I}/A$ and slenderness ratio $\lambda = l/r$. These are calculated when needed.

A1-3 Loadings.

NBEL different loadcases are treated simultaneously in the design process. Because of the optional nonlinear geometric analysis, however, the loadcases are treated sequentially during analysis.

Provisions are made for constant nodal loads, distributed constant and gravity loads on elements and prestressing loads on elements.

In addition to gravity loads another design dependent load, namely an equivalent earthquake loading may be specified.

A1-3.1 Constant nodal loads.

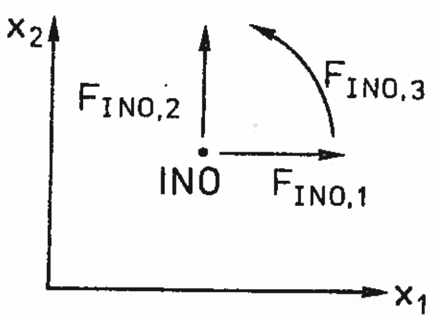


Fig. A1-8 Constant nodal loads.

The constant loads are defined by

$$F_{INO,1} = BNO(IBEL,\ 3 \times INO-2)$$
$$F_{INO,2} = BNO(IBEL,\ 3 \times INO-1) \qquad (A1-14)$$
$$F_{INO,3} = BNO(IBEL,\ 3 \times INO)$$

A1-3.2 Distributed element loads.

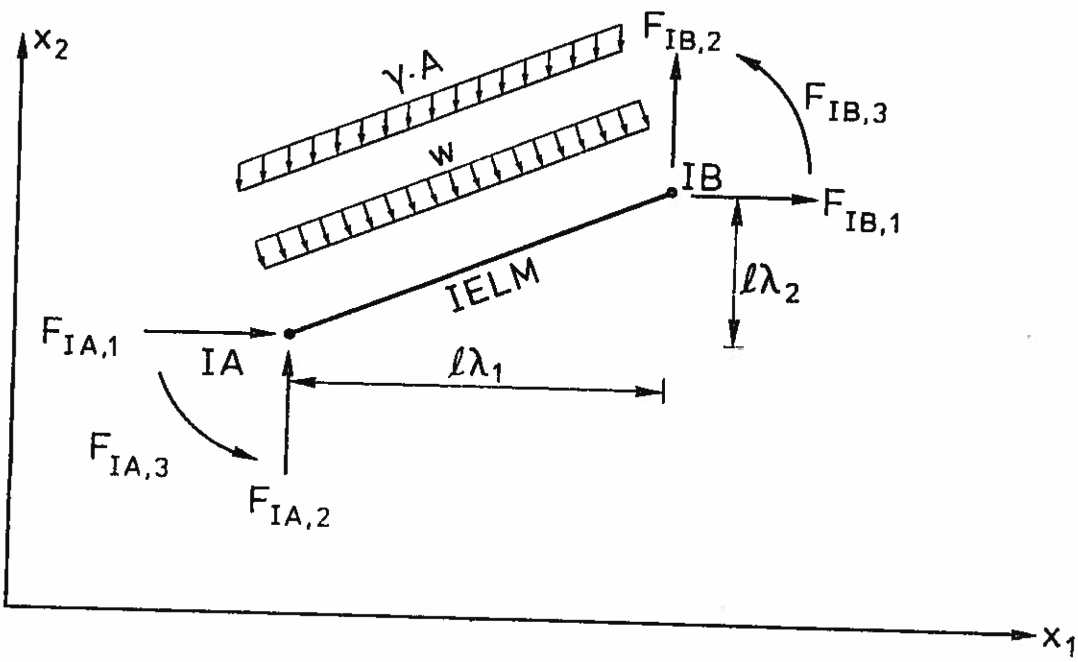


Fig. A1-9 Distributed element loads.

The projection of the gravity load perpendicular to the element is treated as a distributed load, while the projection along the element is treated as concentrated loads acting on the nodes.

The specific gravity, γ, may depend on loadcase and profile

$$\gamma = GAMMA(IBEL,IPROF) \qquad (A1-14)$$

The constant load w may depend on loadcase and element

$$w = PJ(IBEL,IELM) \qquad (A1-15)$$

The gravity and constant loads together define the distributed element load p

$$p = w + \gamma A \cdot \lambda_1 \qquad (A1-16)$$

The action of the distributed element loads in the nodal directions is given in section A1-5.

The gravity load projection along the element gives the following nodal loads

$$F_{IA,1} = F_{IB,1} = -\tfrac{1}{2}\gamma A l \lambda_2 \lambda_1$$

$$F_{IA,2} = F_{IB,2} = -\tfrac{1}{2}\gamma A l \lambda_2^2 \qquad \text{(A1-17)}$$

A1-3.3 The prestressing load.

If desired an automatic prestressing of all pin-pin elements will be executed by specifying decision parameter AUTOVNF = 1. The pin-pin elements are then given a lack of fit, V_{Nf}, yielding half the allowable stress in tension in loadcase IBEL = 1. With this amount of prestress the pin-pin elements will most likely stay in tension under subsequent loadcases.

The contribution of the lack of fit based prestressing force, $N_f = V_{Nf}EA/l$, to the total nodal load is given in section A1-5.

A1-3.4 The equivalent earthquake load.

The equivalent earthquake loading is based on the SEAOC Recommendations, ref. [7]. It consists of lateral nodal loads which depend upon the distribution of the total mass, the total mass itself, the fundamental period and the type of building frame.

This loading is invoked if decision parameter ANSVING = 1. The loading is superposed on loadcase IBEL = 2.

The mass, WM(INO), lumped at node INO is defined as the total vertical load at node INO divided by the acceleration of gravity, $g = 9.81\ m/sec^2$. The height above ground is taken as X2(INO). The fundamental period T, is calculated at the current design with lumped masses, WM, using the iterative method shown in appendix 3.

The loading then consists of lateral nodal loads given by

$$F_{INO,1} = C\cdot\frac{0.05}{\sqrt[3]{T}}\, g \sum_{1,NNO} WM(INO)\, \frac{WM(INO)\cdot X2(INO)}{\sum\limits_{1,NNO} WM(INO)\cdot X2(INO)} \qquad \text{(A1-18)}$$

Here C is a coefficient dependent on the type of building frame. Sway frames have smaller values than braced frames. Values of

C = 1.00 and C = 1.33 have been used in the design examples.

A1-4 Force-displacement relations of the elements.

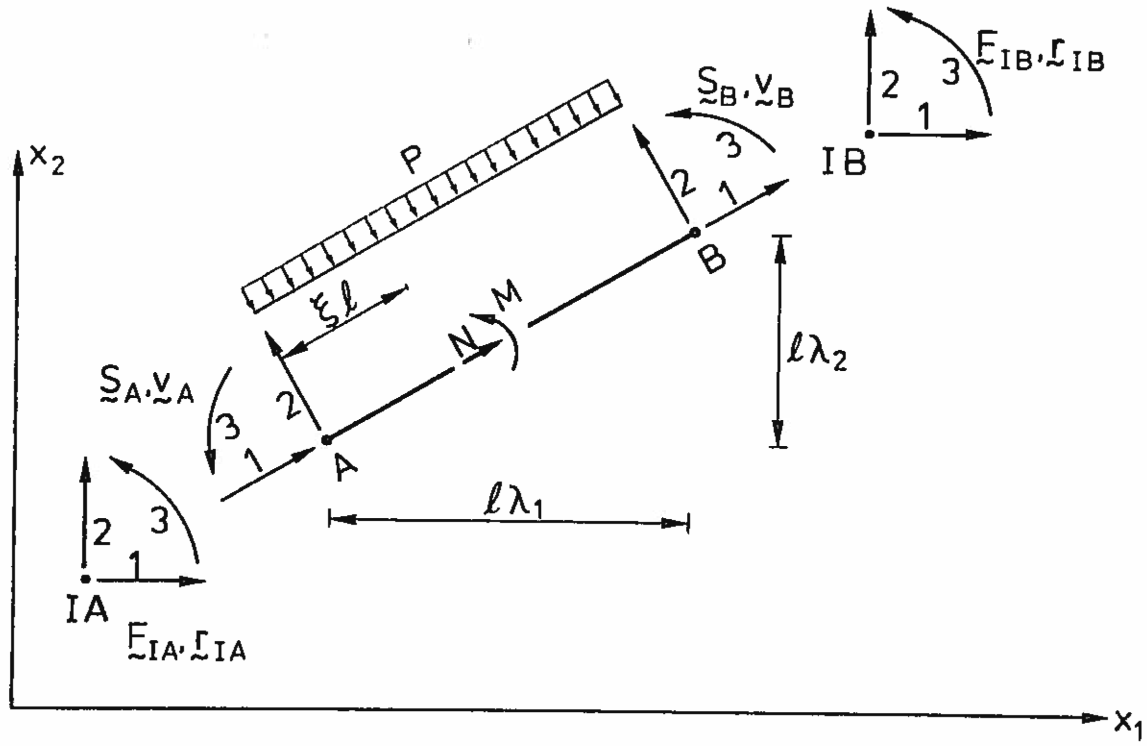


Fig. A1-10 Local and global directions.

Between local and global end displacements and forces simple geometric relations exist.

$$\begin{bmatrix} \underset{\sim}{v}_A \\ \underset{\sim}{v}_B \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\lambda} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\lambda} \end{bmatrix} \begin{bmatrix} \underset{\sim}{r}_{IA} \\ \underset{\sim}{r}_{IB} \end{bmatrix} \tag{A1-19}$$

$$\begin{bmatrix} \underset{\sim}{F}_{IA} \\ \underset{\sim}{F}_{IB} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\lambda}^T & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\lambda}^T \end{bmatrix} \begin{bmatrix} \underset{\sim}{S}_A \\ \underset{\sim}{S}_B \end{bmatrix} \tag{A1-20}$$

where the orthogonal transformation matrix $\underset{\sim}{\lambda}$ is given by

$$\underset{\sim}{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & 0 \\ -\lambda_2 & \lambda_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A1-21}$$

The following force-displacement relations are based on derivations given by Holst, ref. [6]. Only expressions from the consistent nonlinear geometric analysis are stated corresponding to decision parameter ISTAB = 5. Linear geometric analysis is performed when ISTAB = 0, but will not be described here.

A1-4.1 Transversal displacement.

Define the following deformation and stiffness parameters.

$$\begin{aligned} v_F &= (v_{B,2} - v_{A,2})/l \\ v_A &= -v_{A,3} - v_F \\ v_B &= v_{B,3} - v_F \\ v_N &= v_{B,1} - v_{A,1} \end{aligned} \qquad \text{(A1-22)}$$

$$kl = \sqrt{\frac{-N}{EI}}\, l$$

The downward transversal displacement, u_2, at a distance $X = \xi l$ from A is then given by

$$u_2 = C_1 \sin kx/k + C_2 l \cos kx + C_3 x + C_4 l + \tfrac{1}{2} C_0 x^2/l \qquad \text{(A1-23)}$$

where the real, dimensionless constants C_0-C_4 are

$$C_0 = (p + N \cdot x_f)/(EI(kl)^2) \qquad \text{(A1-24)}$$

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \frac{1}{(kl)^2} \begin{bmatrix} (kl)^2 - B_1 & 0 & 0 \\ A_1 & A_2 & 0 \\ B_1 & -B_1 & -(kl)^2 \end{bmatrix} \begin{bmatrix} v_A \\ v_B \\ v_F \end{bmatrix} + \tfrac{1}{2} C_0 \begin{bmatrix} 1 \\ (A_1+A_2)/(kl)^2 \\ 1 \end{bmatrix} \qquad \text{(A1-25)}$$

$$C_4 = -v_{A,2}/l - C_2 \qquad \text{(A1-26)}$$

and

$$\begin{aligned} A_1 &= (kl \sin kl - (kl)^2 \cos kl)/d \\ A_2 &= (kl \sin kl - (kl)^2)/d \\ d &= 2 - 2 \cos kl - kl \sin kl \\ B_1 &= A_1 - A_2 \end{aligned} \qquad \text{(A1-27)}$$

A1-4.2 Section forces.

From the transversal displacement u_2, the moment M is derived

$$\begin{aligned} M &= - EI \cdot u_2'' \\ &= \frac{EI}{l} (C_1\, kl \sin kx + C_2 (kl)^2 \cos kx - C_0) \end{aligned} \qquad \text{(A1-28)}$$

The normal force N, is given by

$$N = \frac{EA}{l} (v_N + v_{Nf} + \delta_G - \delta_{Gf}) \qquad \text{(A1-29)}$$

where v_N and v_{Nf} are already defined and δ_G equals geometric shortening due to sway and curvature and δ_{Gf} is δ_G due to initial curvature.

The expression for δ_G is rather lengthy

$$\begin{aligned}\delta_G = {} & \tfrac{1}{2} l (\tfrac{1}{3} C_0^2 + C_3^2 + C_0 C_3 + \tfrac{1}{2}(C_1^2 + C_2^2 (kl)^2) \\ & + \tfrac{1}{2}(C_1^2 - C_2^2 (kl)^2) \cos kl \sin kl/kl - 2C_0 (C_1 (1-B_1)/(kl)^2 \\ & + C_2 A_1) d/(kl)^2 - C_1 C_2 \sin^2 kl + 2C_3 (C_1 \sin kl/kl + C_2 (\cos kl - 1)) \\ = {} & l (v_F^2/2 + f(v_A, v_B, (p + N x_f) l^3/EI, kl))\end{aligned} \qquad \text{(A1-30)}$$

where deformation parameters are defined different from before

$$\begin{aligned} v_A &= -v_{A,3} + v_F - \tfrac{1}{2} l x_f \\ v_B &= v_{B,3} - v_F - \tfrac{1}{2} l x_f \end{aligned} \qquad \text{(A1-31)}$$

The expression for δ_{Gf} is very simple, due to the constant initial curvature.

$$\begin{aligned}\delta_{Gf} &= \delta_G (v_A = v_B = -\tfrac{1}{2} l x_f,\ v_F = kl = p = N = 0) \\ &= (l^3 x_f^2)/24\end{aligned} \qquad \text{(A1-32)}$$

Fig. A1-11 gives a graphic representation of v_N, v_{Nf} and $\delta_G - \delta_{Gf}$. The heavy linie shows the final displacement of the element with the exception of a fixed body translation.

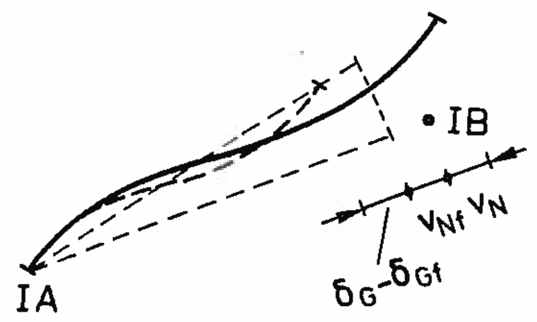


Fig. A1-11 Graphic representation of v_N, v_{Nf} and $\delta_G - \delta_{Gf}$.

A1-5 Element end forces and displacements.

In reference [6] the end force-displacement relations for one element are given in local directions as

$$\begin{bmatrix} \underset{\sim}{S}_A \\ \underset{\sim}{S}_B \end{bmatrix} = \begin{bmatrix} \underset{\sim}{K}_{AA} & \underset{\sim}{K}_{AB} \\ \underset{\sim}{K}_{BA} & \underset{\sim}{K}_{BB} \end{bmatrix} \begin{bmatrix} \underset{\sim}{v}_A \\ \underset{\sim}{v}_B \end{bmatrix} + \begin{bmatrix} \underset{\sim}{S}^o_A \\ \underset{\sim}{S}^o_B \end{bmatrix} \qquad \text{(A1-33)}$$

Applying the previously stated transformations between local and global entities one obtains

$$\begin{bmatrix} \underset{\sim}{F}_{IA} \\ \underset{\sim}{F}_{IB} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\lambda}^T & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\lambda}^T \end{bmatrix} \begin{bmatrix} \underset{\sim}{K}_{AA} & \underset{\sim}{K}_{AB} \\ \underset{\sim}{K}_{AA} & \underset{\sim}{K}_{BB} \end{bmatrix} \begin{bmatrix} \underset{\sim}{\lambda} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\lambda} \end{bmatrix} \begin{bmatrix} \underset{\sim}{r}_{IA} \\ \underset{\sim}{r}_{IB} \end{bmatrix} + \begin{bmatrix} \underset{\sim}{\lambda}^T & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{\lambda}^T \end{bmatrix} \begin{bmatrix} \underset{\sim}{S}^o_A \\ \underset{\sim}{S}^o_B \end{bmatrix}$$

$$= \begin{bmatrix} \underset{\sim}{K}_{IAIA} & \underset{\sim}{K}_{IAIB} \\ \underset{\sim}{K}_{IBIA} & \underset{\sim}{K}_{IBIB} \end{bmatrix} \begin{bmatrix} \underset{\sim}{r}_{IA} \\ \underset{\sim}{r}_{IB} \end{bmatrix} + \begin{bmatrix} \underset{\sim}{F}^o_{IA} \\ \underset{\sim}{F}^o_{IB} \end{bmatrix} \qquad \text{(A1-34)}$$

Writing out the equations for one loadcase gives

$$\begin{bmatrix} F_{IA,1} \\ F_{IA,2} \\ F_{IA,3} \\ F_{IB,1} \\ F_{IB,2} \\ F_{IB,3} \end{bmatrix} = \begin{bmatrix} \lambda_1^2k_1+\lambda_2^2k_2 & \lambda_1\lambda_2(k_1-k_2) & -\lambda_2k_4 & -\lambda_1^2k_1-\lambda_2^2k_2 & \lambda_1\lambda_2(k_2-k_1) & -\lambda_2k_4 \\ & \lambda_1^2k_2+\lambda_2^2k_1 & \lambda_1k_4 & \lambda_1\lambda_2(k_2-k_1) & -\lambda_1^2k_2-\lambda_2^2k_1 & \lambda_1k_4 \\ & & k_3 & \lambda_2k_4 & -\lambda_1k_4 & k_5 \\ & & & \lambda_1^2k_1+\lambda_2^2k_2 & \lambda_1\lambda_2(k_1-k_2) & \lambda_2k_4 \\ & & & & \lambda_1^2k_2+\lambda_2^2k_1 & -\lambda_1k_4 \\ & & & & & k_3 \end{bmatrix} \begin{bmatrix} r_{IA,1} \\ r_{IA,2} \\ r_{IA,3} \\ r_{IB,1} \\ r_{IB,2} \\ r_{IB,3} \end{bmatrix}$$

$$+ \begin{bmatrix} \lambda_1S^o_{A,1} - \lambda_2S^o_{A,2} \\ \lambda_2S^o_{A,1} + \lambda_1S^o_{A,2} \\ S^o_{A,3} \\ \lambda_1S^o_{B,1} - \lambda_2S^o_{B,2} \\ \lambda_2S^o_{B,1} + \lambda_2S^o_{B,2} \\ S^o_{B,3} \end{bmatrix} \qquad \text{(A1-35)}$$

where

$$k_1 = \frac{EA}{l} \qquad k_4 = \frac{EI}{l^2}B_1$$
$$k_2 = \frac{EI}{l^3}(2B_1-(kl)^2) \qquad k_5 = -\frac{EI}{l}A_2 \qquad \text{(A1-36)}$$
$$k_3 = \frac{EI}{l}A_1$$

and

$$S^{o}_{A,1} = -S^{o}_{B,1} = -\frac{EA}{l}(\delta_G - \delta_{Gf} + v_{Nf})$$
$$S^{o}_{A,2} = S^{o}_{B,2} = \tfrac{1}{2}pl \qquad (A1\text{-}37)$$
$$S^{o}_{A,3} = -S^{o}_{B,3} = (p + N \cdot x_f)l^2 \cdot A_3$$

$$A_3 = (2 - (A_1+A_2))/(2(kl)^2) \qquad (A1\text{-}38)$$

The above stated element end force-displacement relations are derived for fix-fix type elements (IKE=1). If the elements are of the pin-pin type (IKE=4), then the same relations hold albeit, with $A_1=A_2=A_3=0$.

A1-6 Force-displacement relations of the total structure.

The preceding loadings and force-displacement relations of element are all stated entirely in global node directions. Looping over all elements, summing up all forces acting in each nodal direction and requiring equilibrium of the forces one obtains the following global force-displacement relations of the total structure.

$$\underset{\sim}{F}(\underset{\sim}{x}) = \underset{\sim}{K}(\underset{\sim}{x})\cdot\underset{\sim}{x} \qquad (A1\text{-}39)$$

where $\underset{\sim}{F}$ is the global loading vector (including the F^o contributions), $\underset{\sim}{K}$ the global stiffness matrix and $\underset{\sim}{x}$ the global displacement vector.

The equations are nonlinear and solved by the simple iterative process

$$\underset{\sim}{F}(\underset{\sim}{x}^{(n-1)}) = \underset{\sim}{K}(\underset{\sim}{x}^{(n-1)})\cdot\underset{\sim}{x}^{(n)} \qquad (A1\text{-}40)$$

Convergence defined by $|\underset{\sim}{x}^{(n)}-\underset{\sim}{x}^{(n-1)}| \leq |\underset{\sim}{x}^{(n-1)}|/100$ is rapidly achieved. (2 or 3 iterations for normal building frames).

Since $\underset{\sim}{K}$ depends on $\underset{\sim}{x}$, only one loadcase may be solved at a time.

The actual solution of the linear equations in the iterative process takes advantage of the symmetric band nature of $\underset{\sim}{K}$ using a banded matrix Cholesky decomposition scheme, explicitly bypassing all equations belonging to removed degrees of freedom (UST(IBEL,IDIS)=0).

The $\underset{\sim}{F}$ vector is generated in procedure BEL.

In case of earthquake loads the natural period, T, is calculated in procedure SVINGAN(cfr. appendix 3) and the $\underset{\sim}{F}$ vector is augmented in procedure EQUBE.

The $\underset{\sim}{K}$ matrix is generated using procedures KEMAT and KEIKK, while decomposition and back substitution is done in procedures MFSB and MTDSBX respectively.

A1-7 Resulting element forces.

Once the global displacements are found the local element displacements and deformation parameters are given by formulas (A1-19) and (A1-22).

While the definition of deformation parameters v_F and v_N applies both to fix-fix and pin-pin type elements, the definition (A1-22) of v_A and v_B applies only to fix-fix type elements. In pin-pin type elements parameters v_A and v_B are determined explicitly in terms of the stiffness parameter kl, loading p and if the element is in compression also the end eccentricity, initial curvature and the normal force.

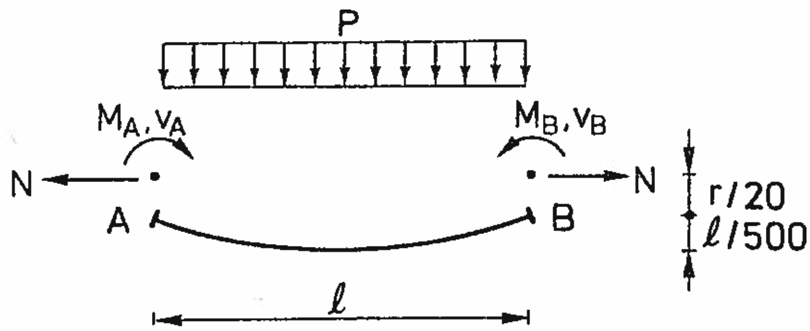


Fig. A1-12 Pin-pin type element with end eccentricity and initial curvature $\varkappa_f$.

Using the element end moment deformation relation (AIII31) of ref. [6] and imposing symmetry (v_A=v_B & M_A=M_B) one finds

$$v_A = \frac{(p + N\varkappa_f)l^2A_3 + M_A}{\mu(A_1 + A_2)} \qquad \text{(A1-41)}$$

In tension $x_f = 0$ and $M_A = 0$ by definition which yields

$$v_A = v_B = \frac{pl^2 A_3}{\mu(A_1 + A_2)} \quad \text{(A1-42)}$$

In compression with initial constant curvature x_f with max displacement equal to l/500, and $M_A = -Nr/20$ from end eccentricity equal to r/20 one finds

$$v_A = v_B = \frac{(p + Nx_f)l^2 A_3 - N\frac{r}{20}}{\mu(A_1 + A_2)} \quad \text{(A1-43)}$$

The sign of the initial curvature and the eccentricity is reversed if p is negative.

The deformation parameters now being fully defined for both fix-fix and pin-pin elements, one finds the desired moment and normal force in the element using the formulas of section A1-4.2.

Appendix 2.

Moments of inertia and section moduli for geometrically similar profiles and available profile series.

In the case of geometrically similar (GS) profiles, i.e., profiles where change of area does not change the shape of the profile, the following two formulas apply

$$I = \alpha A^{2.0} = \alpha \cdot 10^{-4}(100A)^{2.0} \qquad \text{(A2-1)}$$

$$W = \beta A^{1.5} = \beta \cdot 10^{-3}(100A)^{1.5} \qquad \text{(A2-2)}$$

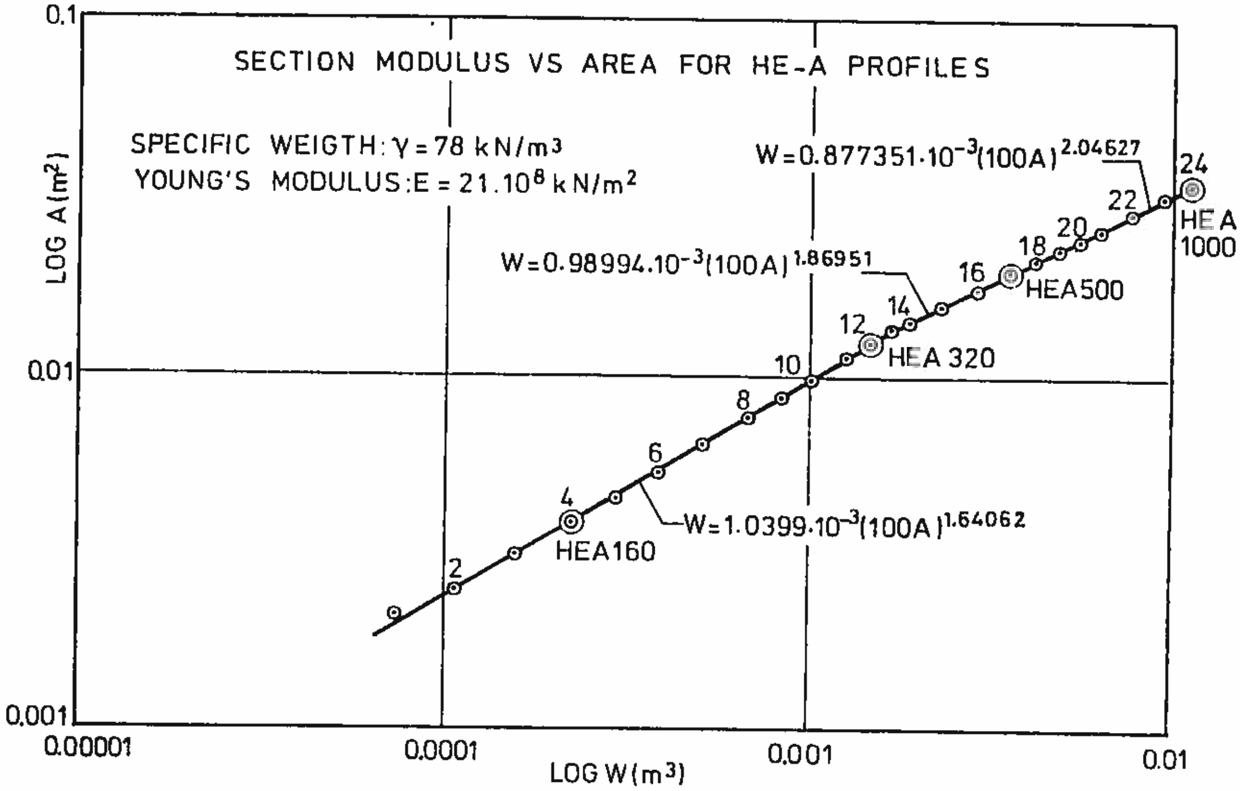
I being the moment of inertia, W the section modulus, A the area and α,β dimensionless constants for the particular profile.

The values of α,β for a few characteristic GS profiles are given below

GS profile series	α	β/α
Solid square	$\frac{1}{12}$	2
Circular ring, ω = thickness/mean radius	$\frac{1+\frac{1}{4}\omega^2}{4\pi\omega}$	$\frac{\sqrt{2\pi\omega}}{1+\frac{1}{2}\omega}$
Solid circle = circular ring with ω = 2	$\frac{1}{4\pi}$	$\sqrt{\pi}$

Table A2-1.

Regarding available profile series formulas of the same nature apply. In order to illustrate this a number of EURONORM profile series have been plotted in log I vs. Log A and log W vs. log A diagrams. In this study the IPE, the HE-A and the HE-B profile series are used. The diagrams in fig. A2-1 show how the empirically obtained power functions (line segments in the log log diagram) fit the HE-A profiles. Diagrams for the other profile series are alike and not shown here. The line segments are determined by the visually best fit and forced to pass through certain profiles thus facilitating ease of calculation of the formulas.



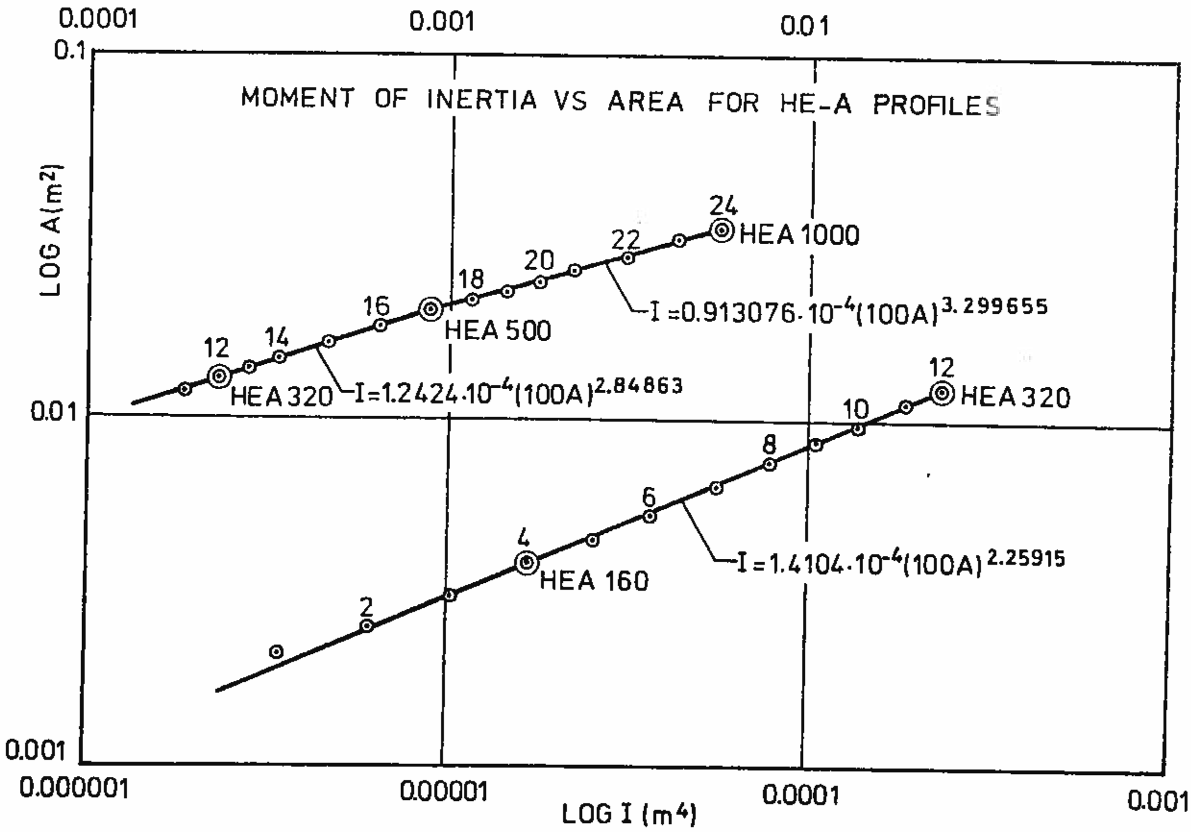


Fig. A2-1. Log A vs. log I and log A vs. log W.

Since the formulas contain dimensional constants it should be remembered that meter is the chosen dimension of length.
For calculation purposes it is desirable that the radix in the formulas has the order of magnitude 1, hence for the above mentioned profile series a scaling factor of 100 is chosen.

In general the formulas are described by

$$A_1 < A < A_2: \quad I = F_I(100A)^{E_I} \qquad \text{(A2-3)}$$

$$W = F_W(100A)^{E_W} \qquad \text{(A2-4)}$$

For the IPE, the HE-A, the HE-B and the GS series the following values apply.

IPROF	Profile Series	$100A_1$ / $100A_2$	$F_I \cdot 10^4$	E_I	$F_W \cdot 10^3$	E_W
1	IPE	0.0764				
			3.42084	2.31026	1.50986	1.65410
		1.5600				
2	HE-A	0.212				
			1.41040	2.25915	1.03990	1.64062
		1.240				
			1.24245	2.84863	0.98994	1.86951
		1.980				
			0.91308	3.29956	0.87735	2.04627
		3.470				
3	HE-B	0.26				
			1.02348	2.31478	0.86732	1.67956
		1.61				
			0.63302	3.32365	0.71585	2.08261
		4.00				
4	GS	$100A_{min}$				
			α	2.00000	β	1.50000
		$100A_{max}$				

Table A2-2

Notice the difference in the exponents between the Geometrically Similar series and the other series. Both E_I and E_W are greater in the available series which gives greater increase in stiffness for a given increase in area. Geometrically similar series are seldom if ever used because of their poor economy.

The described relationships are implemented in procedure PROFIL.

Appendix 3.

Solution of the eigenproblem of a n dimensional discrete system by an iterative method.

The oscillations of a discrete system can be described by the eigenvalue problem

$$(\underset{\sim}{K} - \frac{1}{\lambda_j}\underset{\sim}{M})\cdot\underset{\sim}{v}_j = \underset{\sim}{O} \qquad \text{(A3-1)}$$

where $\underset{\sim}{K}$ is the global stiffness matrix (may depend on the normal force distribution), $\underset{\sim}{M}$ is the inertia or mass matrix, $\underset{\sim}{v}_j$ is the oscillationvector of the j'th eigenmode and $\lambda_j = (T_j/2\pi)^2$ is the associated j'th eigenvalue and T_j the period of the j'th mode oscillations.

Provided that the matrix p norm $||\underset{\sim}{K}^{-1}\cdot\underset{\sim}{M}||_p < 1$ and $|\lambda_1|>|\lambda_2|>\ldots$ $\ldots>|\lambda_m|>|\lambda_{m+1}|\geq\ldots\geq|\lambda_n|$ the m largest eigenmodes can be determined by the following iterative method (here stated without proof).

Starting with $m \leq n$ linear independent approximations to the eigenvectors the iteration reads

$$\underset{\sim}{u}_j = \underset{\sim}{K}^{-1}\cdot\underset{\sim}{M}\cdot\underset{\sim}{v}_j^{(i)} \qquad (j = 1,m) \qquad \text{(A3-2)}$$

$$\underset{\sim}{z}_j = \underset{\sim}{u}_j - \sum_{k=1}^{j-1}\frac{\underset{\sim}{u}_j\cdot\underset{\sim}{M}\cdot\underset{\sim}{z}_k}{\underset{\sim}{z}_k\cdot\underset{\sim}{M}\cdot\underset{\sim}{z}_k}\,\underset{\sim}{z}_k \qquad (j = 1,m) \qquad \text{(A3-3)}$$

$$\lambda_j^{(i+1)} = ||\underset{\sim}{z}_j||_p \qquad (j = 1,m) \qquad \text{(A3-4)}$$

$$\underset{\sim}{v}_j^{(i+1)} = \underset{\sim}{z}_j/\lambda_j^{(i+1)} \qquad (j = 1,m) \qquad \text{(A3-5)}$$

This method is an extension of the wellknown power method for the largest eigenmode. It is based on the $\underset{\sim}{M}$ orthogonality of the eigenvectors and requires $\underset{\sim}{K}$ and $\underset{\sim}{M}$ to be symmetric.

The $\underset{\sim}{u}_j$ in equation (A3-2) is solved from the equations

$$\underset{\sim}{K}\cdot\underset{\sim}{u}_j = \underset{\sim}{M}\cdot\underset{\sim}{v}_j^{(i)} \qquad (j = 1,m)$$

using the Cholesky method and solving for all m vectors simultaneously.

In equation (A3-4) the p = 1 norm is used because of simplicity.

$$\lambda_j^{(i+1)} = ||\underset{\sim}{z}_j||_p = (\sum_{i=1}^{n} |z_{ji}|^p)^{\frac{1}{p}}$$

$$= \sum_{i=1}^{n} |z_{ji}| \text{ with } p = 1 \qquad \text{(A3-6)}$$

Iterations are continued until a specified relative convergence of the eigenvalues

$$|\lambda_j^{(i+1)} - \lambda_j^{(i)}| < eps.|\lambda_j^{(i)}| \qquad (j=1,m) \qquad \text{(A3-7)}$$

With fair approximations to the eigenvectors this method converges very rapidly.

In iterative design the eigenvectors of the previous step will always be good approximations to the current eigenvectors and thus justify the use of the above described iterative eigenproblem solution.

In procedure SVINGAN the above method is implemented for a lumped mass system. The mass matrix in a lumped mass system is diagonal and for computational reasons the mass matrix is stored 1 dimensionally.

Appendix 4. Program listings.

In this appendix all procedures are listed. In order to minimize the computer main storage requirements the program is fitted into an overlay structure which is also listed.

Most of the procedures are selfexplanatory and/or explained in chapter 4 or the first three appendices.

Some of the procedures are completely general and can be used outside of the present context.

DESIGN is the main procedure from which all other procedures directly or indirectly are invoked.

Since some of the calculations in the program require great accuracy and the datamachine used is IBM 370/165 all reals are stored in double prescision and consequently all calculations involving reals are carried out in double prescision.

Typical resource demands for the three design examples of the main report are shown below.

FRAME	NELM	NBEL	NDIS	NBÅND	MAIN CORE (Kbytes)	LIN. ANAL. NAGRUP	NITERA	CPU (Sec)	NONLIN. ANAL. NAGRUP	NITERA	CPU (Sec)
OH-2E1F	6	1	18	8	104	4	5	3.8	4	3	5.0
OH-2E1FU	8	3	24	11	106	4	8	8.2	3	2	5.9
OH-10E3F	70	3	132	14	144	12	8	73	4	3	44
OH-10E3FG	90	3	132	17	152	15	8	175	11	3	107
OH-30E2F	150	2	279	11	174	15	8	120	4	3	51

Table A4-1. Problems size and resource demand on IBM Model 370/165.

```
STMT LEVEL NEST
  1                  ANALYS:      /* STATIC AND DYNAMIC ANALYSIS OF A PLANE FRAME                          */
                                  /* LINEAR GEOMETRIC IF ISTAB=0,ELSE  NONLINEAR GEOMETRIC                 */
                     PROCEDURE(ELEMTOPO,ELEMGEO,X2,MILIEU,C,STIVHED,KK,KE,F,RESPONS,EV,T,W);
  2      1           DECLARE ((NELM,DOFNO,NDIS,NBEL,NBAND,ISTAB,NNC,NOPEL,AUTOVNF,
                             NEV,ANSVING,REANAL) EXTERNAL,
                             (IELM,IENS,IDIS,IDEL,IT,IGRUP,NYIT)INITIAL(1)) BIN FIXED,
                             ERROR CHAR(1) EXTERNAL,
                             (W,C,(X2,KE,F,T)(*),WM(NNO),MASSE(NDIS),
                             (KK,EV)(*,*)) BIN FLOAT(53),
                             1 ELEMTOPO,
                               2 ((LGRUP,LENS,LKE,LNSNIT,LPROF)(*),TOPO(*,*)) BIN FIXED,
                             1 ELEMGEO,        2 (SL(*),A(*,*),VNF(*),EPSF) BIN(53),
                             1 MILIEU(*),
                               2 BELASTNING,3 (PJ,ENC,GAMMA)(*) BIN(53),
                               2 FLYTNING,    3 (LNODIS(*) FIXED,UST(*) FLOAT(53)) BIN,
                             1 STIVHED,       2 ((AREA,IMOM)(*),EMOD) BIN(53),
                             1 RESPONS(*),    2 (X,KLA,DKA)(*) BIN(53);
  3      1           IF ANSVING=1 THEN
  4      1           CALL MASS(MILIEU.BELASTNING(2),ELEMTOPO,ELEMGEO,AREA,MASSE,WM);
  5      1           DO IBEL=1 TO NBEL;          NYIT=1;
  7      1     1     DO IT=1 TO 15 WHILE(NYIT=1);                                  NYIT=0;
  9      1     2     IF ISTAB=0 & IBEL-AUTOVNF>1 THEN GOTO BELASTNING;
 11      1     2     KK=0E0;
                                  /* BUILD UP GLOBAL STIFFN.MATRIX                                         */
 12      1     2     DO IELM=1 TO NELM;
 13      1     3     IF AUTOVNF=1 & IBEL=1 & LKE(IELM)=4 THEN GOTO INDEND;
 15      1     3     IENS=LENS(IELM);              IGRUP=LGRUP(IELM);
 17      1     3     CALL KEMAT(SL(IENS),A(IENS,*),AREA(IGRUP),IMOM(IGRUP),EMOD,
                                  KLA(IBEL,IELM),LKE(IELM),KE);
 18      1     3     CALL KEIKK(KK,KE,TOPO(IELM,*),NOPEL,DOFNO);
 19      1     3     INDEND:END;
                                  /*  REVISE KK IF SUPPORT OR INTERNAL PIN CONNECTION                      */
 20      1     2     DO IDIS=1 TO NDIS;
 21      1     3     IF UST(IBEL,IDIS) =0E0 THEN KK(IDIS,1)=1E20;    END;
                                  /* DECOMPOSE KK
 24      1     2     CALL MFSB(KK,NDIS,NBAND);                                                     */
 25      1     2     BELASTNING:
                     CALL BEL(ELEMTOPO,ELEMGEO,MILIEU.BELASTNING(IBEL),STIVHED,F,
                              RESPONS(IBEL),IBEL);
 26      1     2     IF ANSVING=1 THEN IF IBEL=2 THEN DO;
 29      1     3     CALL SVINGAN(KK,MASSE,EV,T,NDIS,NBAND,NEV);
 30      1     3     CALL EQUBE(UST(2,*),X2,WM,W,C,T(1),X(2,*));
 31      1     3     PUT EDIT('T',T)(SKIP,A,(NEV)F(12,3));
 32      1     3     END;
 33      1     2     IF ERROR='0' THEN
 34      1     2     CALL MTDSBX(KK,X,NDIS,NBAND,IBEL,IBEL);
 35      1     2     ELSE RETURN;
                                  /* CHECK CONVERGENCE IF ISTAB>0
 36      1     2     CALL KONVERG(MILIEU.BELASTNING(IBEL),ELEMTOPO,ELEMGEO,STIVHED,            */
                                  RESPONS(IBEL),NYIT,IBEL);
 37      1     2     IF REANAL=1 THEN NYIT=0;
 39      1     2     END;
 40      1     1     END;
 41      1           END ANALYS;
```

```
BEL:        /* TOTAL NODAL FORCES ARE COMPUTED AND STORED IN X          */BEL00010                PAGE

STMT LEVEL NEST
  1                 BEL:        /* TOTAL NODAL FORCES ARE COMPUTED AND STORED IN X
  2     1           PROCEDURE(ELEMTOPC,ELEMGEO,MILIEUB,STIVHED,F,RESPONS,IBEL);              */BEL00010
                    DECLARE ((NELM,NNC,DOFNC,NKRAFT,AUTOVNF) EXTERNAL,                         BEL00020
                            IELM,IENS,IGRUP,IKE,IA,IB,IDIS,IBEL,IPROF) BIN FIXED,              BEL00030
                            (HORZBEL EXTERNAL,                                                 BEL00040
                            KL,FA,FB,COSFI,SINFI,L,AREAL,B1,B2,B3,B4,B5,P,EP,                  BEL00050
                            F(*)) BIN FLOAT(53),                                               BEL00060
                            1 ELEMTOPC,                                                        BEL00070
                              2 ((LGRUP,LENS,LKE,LNSNIT,LPROF)(*),TOPC(*,*)) BIN FIXED,        BEL00080
                            1 ELEMGEO,      2 (SL(*),A(*,*),VNF(*),EPSF) BIN(53),              BEL00090
                            1 MILIEUB,      2 (PJ,BNO,GAMMA)(*) BIN FLOAT(53),                 BEL00100
                            1 STIVHED,      2 ((AREA,IMOM)(*),EMOD) BIN FLOAT(53),             BEL00110
                            1 RESPONS,      2 (BNOEFF,KLA,DGA)(*) BIN FLOAT(53);               BEL00120
  3     1           B4,B5=0E0;                                                                 BEL00130
  4     1           BNOEFF=BNO;                                                                BEL00140
  5     1           DO IELM=1 TO NELM;                                                         BEL00150
  8     1     1     IA=DOFNO*TOPO(IELM,1);  IENS=LENS(IELM);        IGRUP=LGRUP(IELM);         BEL00160
 11     1     1     COSFI=A(IENS,1);        IB=DOFNC*TOPO(IELM,2);  IKE=LKE(IELM);             BEL00170
 14     1     1     AREAL=AREA(IGRUP);      SINFI=A(IENS,2);        IPROF=LPROF(IGRUP);        BEL00180
 16     1     1     P=PJ(IELM);             L=SL(IENS);                                        BEL00190
 18     1     1     B1=P*SINFI/2E0;         EP=GAMMA(IPROF)*AREAL*L;                           BEL00200
 21     1     1     KL=KLA(IELM);           B2=-(P*COSFI+EP)/2E0;   B3=(P+EP*COSFI)*L;         BEL00210
 22     1     1     IF NKRAFT=1 & ABS(KL)>.15 & IKE=1 THEN                                     BEL00220
 23     1     1     CALL KLVIRK (KL,FA,FE,'BEL');         ELSE                                 BEL00230
 24     1     1     DO;                     FA=F(IKE);                                         BEL00240
                                                            FB=F(IKE+4);            END;       BEL00250
 28     1     1            /* AUTOMATIC PRESTRESS OF BRACING ELEMENTS                         */BEL00260
                    IF AUTOVNF=1 & IBEL=1 & IKE=4 THEN  P=EPSF*EMOD*AREAL; ELSE                BEL00270
                           /* GEOMETRICAL SHORTENING,DGA OR -PRESTRESS,VNF                    */BEL00280
 30     1     1     P=(DGA(IELM)+VNF(IELM))*EMOD*AREAL/L;                                      BEL00290
 31     1     1     B4=P*COSFI;             B5=P*SINFI;                                        BEL00300
 33     1     1     BNOEFF(IA-2)=BNOEFF(IA-2)+B1+B4;  BNOEFF(IB-2)=BNOEFF(IB-2)+B1-B4;         BEL00310
 35     1     1     BNOEFF(IA-1)=BNOEFF(IA-1)+B2+B5;  BNOEFF(IB-1)=BNOEFF(IB-1)+B2-B5;         BEL00320
 37     1     1     BNOEFF(IA  )=BNOEFF(IA  )-B3*FA;  BNOEFF(IB  )=BNOEFF(IB  )+B3*FB;         BEL00330
 39     1     1     END;                                                                       BEL00340
 40     1           IF HORZBEL¬=0E0 & IBEL=2 THEN                                              BEL00350
 41     1           DO IDIS=1 BY 3 TO NDIS;                                                    BEL00360
 42     1     1     BNOEFF(IDIS)=BNOEFF(IDIS)+BNOEFF(IDIS+1)*HORZBEL;END;                      BEL00370
 44     1           END BEL;                                                                   BEL00380
```

```
STMT LEVEL NEST
  1              DESIGN:     /* MINIMUM VOLUME DESIGN OF PLANE FRAMES,BY OLE HOLST           */DESIGN10
                 PROCEDURE OPTIONS(MAIN);                                                       DESIGN20
                         /* DECLARATION OF VARIABLES USED IN ANALYSIS                         */DESIGN30
  2     1        DECLARE (HOVED CHAR(100),ERROR CHAR(1)) EXTERNAL,                              DESIGN40
                         ((AUTOPS,AUTCGE,AUTOBE,ANSVING,MNUD,FIUD,REANAL,MAL,FORSTUR,           DESIGN50
                         AUTOVNF,KKRAFT,KOFDE,NLKRUM,ISTAB,NENS)    INITIAL(0),                 DESIGN60
                         (NNO,NELM,NBEL,NDIS,NGRUP,NEV,NVERT,NHORZ,                             DESIGN70
                         NBAND)INIT(1),NSNIT INIT(3),NPROF INIT(4),DOFNO INIT(3),               DESIGN80
                         NOPEL INIT(2), DOFEL INIT(6)) BIN FIXED EXTERNAL,                      DESIGN90
                         (ALPHA INIT(.0795775),BETA INIT(.1410474),HORZEEL INIT(0))             DESIG100
                         BIN FLOAT(53) EXTERNAL;                                                DESIG110
                         /* DECLARATION OF VARIABLES USED IN ITERATIVE REDESIGN               */DESIG120
  3     1        DECLARE ((NFSI,NAGRUP) INITIAL(0),                                             DESIG130
                         (NITERA,NREST,NAREST,NCDIS) INITIAL(1)) BIN FIXED EXTERNAL,            DESIG140
                         (RESTEPS INIT(.2),DELTAA INIT(.0001),RELAXF,RELAXF,                    DESIG150
                         REDUKF,REDUKR,FFM) BIN FLOAT(53) EXTERNAL;                             DESIG160
  4     1        ON ENDPAGE(SYSPRINT) BEGIN;                                                    DESIG170
  6     2        PUT PAGE EDIT(HOVED,'ISTAB=',ISTAB,'NAGRUP=',NAGRUP,'NFSI=',NFSI,              DESIG180
                         'RELAXF=',RELAXF,'REDUKF=',REDUKF,'FFM=',FFM)                          DESIG190
                         (A,SKIP,3 (X(1),A,F(2)),3 (X(1),A,F(5,2)));                            DESIG200
  7     2        PUT SKIP(2);                                                                   DESIG210
  8     2        END;                                                                           DESIG220
  9     1        CALL LÆS;                                                                      DESIG230
 10     1        PUT DATA;                                                                      DESIG240
 11     1        BEGIN;                                                                         DESIG250
                         /* DECLARATION OF VARIABLES USED IN ANALYSIS                         */DESIG260
 12     2        DECLARE 1 ELEMTOPC,                                                            DESIG270
                           2 ((LGRUP,LENS,LKE,LNSNIT)(NELM),LPROF(NGRUP)) BIN FIXED,            DESIG280
                           2 TOPO(NELM,NOPEL) BIN FIXED,                                        DESIG290
                         (X1,X2)(NNO) BIN FLOAT(53);                                            DESIG300
 13     2        CALL LÆS2(ELEMTOPC,X1,X2);                                                     DESIG310
 14     2        BEGIN;                                                                         DESIG320
                         /* DECLARATION OF VARIABLES USED IN ANALYSIS                         */DESIG330
 15     3        DECLARE (T(NEV) INIT((NEV) 1),EV(NEV,NDIS) INIT((NEV*NDIS) 0),                 DESIG340
                         C) BIN FLOAT(53),                                                      DESIG350
                         IBEL BIN FIXED,                                                        DESIG360
                         1 MILIEU(NBEL),                                                        DESIG370
                           2 BELASTNING,                                                        DESIG380
                             3 (PJ(NELM),BNO(NDIS),GAMMA(NPROF)) BIN FLOAT(53),                 DESIG390
                           2 FLYTNING,                                                          DESIG400
                             3 (LNCDIS(NCDIS) FIXED,UST(NCDIS) FLOAT(53)) BIN,                  DESIG410
                         1 ELEMGEO,                                                             DESIG420
                           2 (SL(NENS) INIT((NENS) 0),A(NENS,2)) BIN FLOAT(53),                 DESIG430
                           2 (VNF(NELM),EPSF) BIN FLOAT(53),                                    DESIG440
                         1 STIVHED,                                                             DESIG450
                           2 (AREA,IMOM)(NGRUP) BIN FLOAT(53),                                  DESIG460
                           2 EMOD BIN FLOAT(53),                                                DESIG470
                         1 STYRKE,                                                              DESIG480
                           2 WMOM(NGRUP) BIN FLOAT(53),                                         DESIG490
                           2 (SIGMAN,SIGMAB,LIG)(NBEL,NPROF) BIN FLOAT(53),                     DESIG500
                           2 (FL,FC,FF,EULERN) BIN FLOAT(53),                                   DESIG510
                         1 RESPONS(NBEL),                                                       DESIG520
                           2 X(NDIS) BIN FLOAT(53),                                             DESIG530
                           2 (KLA,CGA)(NELM) INIT((NBEL*NELM) 0) BIN FLOAT(53);                 DESIG540
                         /* DECLARATION OF VARIABLES USED IN ITERATIVE REDESIGN               */DESIG550
 16     3        DECLARE (IAGRUP,IAREST,IR,ITERA,JGL,JNY,LAGRUP(NGRUP)) BIN FIXED,              DESIG560
```

```
STMT LEVEL NEST

 17    3              (AMIN,AMAX,BL,BU,DA,GRAD,FM)(NGRUP)  BIN FLOAT(53);
                CALL LÆS3(ELEMTOPO,ELEMGEO,X1,X2,MILIEU,C,STYRKE,EMUD,EV,           DESIG570
                        AREA,AMIN,AMAX,FM,LAGRUP,GRAD);                             DESIG580
 18    3        CALL PROFIL(AREA,IMOM,WMOM,LPROF,JNY,JGL);                          DESIG590
 19    3        CALL SKRIV3(ELEMTOPO,ELEMGEO,X1,X2,MILIEU,C,STYRKE,STIVHED);        DESIG600
 20    3        IF MAL>0 & FORSTOR=0 THEN      /* PLOT STRUCTURE ON CALCOMP 563     DESIG610
 21    3        CALL PLOTPR(MAL,ELEMTOPO,ELEMGEO,X1,X2);                          */DESIG620
 22    3        BEGIN;                                                              DESIG630
 23    4              /* DECLARATION OF VARIABLES USED IN ANALYSIS                  DESIG640
                DECLARE (ONE INIT(1),TWO INIT(2),                                 */DESIG650
                        KK(NDIS,1:NBAND),KE(21),F(8)) BIN FLOAT(53),                DESIG660
                        /* DECLARATION OF VARIABLES USED IN ITERATIVE REDESIGN      DESIG670
                        (S,VS,FIMAX,W) BIN FLOAT(53),                             */DESIG680
                        SKEMA(NGRUP+4,0:NITERA) BIN FLOAT(53),                      DESIG690
                        1 TABLEAU,                                                  DESIG700
                          2 (NC,NR,TOM INITIAL(0)) BIN FIXED,                       DESIG710
                          2 (IBASIS(1:NREST+NGRUP),NBASIS(1:NGRUP)) BIN FIXED,      DESIG720
                          2 AK(1:NREST+NGRUP,1:NGRUP) BIN FLOAT(53);                DESIG730
 24    4              /* DECLARATION OF VARIABLES USED IN CONSTRAINT EVALUATION     DESIG740
                DECLARE 1 FILISTER,                                               */DESIG750
                          2 (IAR,(LCDIS,LELM,LDEL)(NREST) INIT((NREST) 0)) BIN FIXED, DESIG760
                          2 (FIS,(FI,LKSI)(NREST)) BIN FLOAT(53)  ;                 DESIG770
 25    4        F=0E0;                F(1),F(5)=CNE/12E0;      F(3),F(6)=CNE/8E0;   DESIG780
 28    4        DA=AREA;              AREA,BL,BU=0E0;                               DESIG790
 30    4              /* ITERATIVE REDESIGN LOOP                                    DESIG800
 31    4     1  DO ITERA=0 TO NITERA;                                             */DESIG810
                NYANALYSE:            REANAL=0;                 FIUD=1;             DESIG820
 33    4     1        /* CURRENT DESIGN                                             DESIG830
                AREA=AREA+DA;                                                     */DESIG840
 34    4     1        /* ANALYSIS OF CURRENT DESIGN                                 DESIG850
 35    4     1  CALL PROFIL(AREA,IMOM,WMOM,LPROF,JNY,JGL);                        */DESIG860
                CALL ANALYS(ELEMTOPO,ELEMGEO,X2,MILIEU,C,STIVHED,KK,KE,F,RESPONS,   DESIG870
                        EV,T,W);                                                    DESIG880
 36    4     1        /* EVALUATION OF STRUCTURAL CONSTRAINTS & SELECTION OF        DESIG890
 37    4     1        /* THE SET OF CRITICAL CONSTRAINTS                          */DESIG900
 38    4     1  CALL EVALFI(ELEMTOPO,ELEMGEO,MILIEU,STIVHED,RESPONS,STYRKE,FILISTER); */DESIG910
 41    4     2  S,VS=0E0;                                                           DESIG920
 42    4     2  DO IELM=1 TO NELM;      IGRUP=LGRUP(IELM);      IENS=LENS(IELM);    DESIG930
                S=S+AREA(IGRUP)*SL(IENS);                                           DESIG940
                VS=VS+DA(IGRUP)*SL(IENS);                                           DESIG950
 44    4     1        /* UPDATE AND PRINT THE DESIGN ITERATION REPORT   END;        DESIG960
 45    4     1  CALL OPDAT(FIMAX,FI,SKEMA,ITERA,STIVHED,DA,BL,BU,S,T(1),W,C,FM);  */DESIG970
                CALL SKRIV6(SKEMA,ITERA);                                           DESIG980
 46    4     1        /* CHECK CONVERGENCE & FEASIBILITY                            DESIG990
                IF (ABS(VS)<1.E-3*S)&(FIMAX<=1.002E0) THEN GO TO ITERAEND;        */DESI1000
 48    4     1        /* DECIDE TYPE OF REDESIGN AND COMPUTE REDESIGN BOUNDS        DESI1010
 49    4     1  IF ITERA<NFSI-1 THEN                                              */DESI1020
 52    4     1  DO IGRUP=1 TO NGRUP;    FM(IGRUP)=RELAXF*(FI(IGRUP)-1E0);      END; DESI1030
 54    4     1  IF ITERA=NFSI-1 THEN FM=SQRT(FIMAX)-1E0;                            DESI1040
 55    4     1  IF ITERA=NFSI THEN FM=FFM;                                          DESI1050
 61    4     1  IF ITERA>NFSI THEN DO;    FM=REDUKF*FM;   RESTEPS=REDUKR*RESTEPS; END; DESI1060
 62    4     1  BL=MAX(-ABS(FM)*AREA,AMIN-AREA);                                    DESI1070
 63    4     1  BU=MIN( ABS(FM)*AREA,AMAX-AREA);                                    DESI1080
 65    4     2  IF ERROR='S' THEN DO;                                               DESI1090
                PUT SKIP DATA(ERROR);   DA=BU;                  KLA=0E0;            DESI1100
                                                                                    DESI1110
```

```
DESIGN:      /* MINIMUM VOLUME DESIGN OF PLANE FRAMES,BY OLE FOLST      */DESIGN10

STMT LEVEL NEST

 63    4    2     ITERA=ITERA+1;            GOTO NYANALYSE;                END;                    DESI1120
 71    4    1     IF ITERA<NFSI THEN DO;    /* RELAXED FSD OR FIMAX SCALING                         */DESI1130
 73    4    2     DA=FI*AREA;               DA=MIN(DA,BU);                 DA=MAX(BL,DA);           DESI1140
 76    4    2     ITERA=ITERA+1;            GOTO NYANALYSE;                END;                     DESI1150
                           /* MINIMUM VOLUME REDESIGN BY SEQUENTIAL LINEAR PROGRAM.                 */DESI1160
                           /* CALCULATION OF GRADIENTS OF CRITICAL STRUCTURAL                       */DESI1170
                           /* CONSTRAINTS BY A BACKWARD DIFFERENCE SCHEME                           */DESI1180
 79    4    1     REANAL=1;                 JNY,FIUD,MNUD=0;               AK=0E0;                  DESI1190
                           /* LOOP OVER ACTIVE DESIGN VARIABLES                                     */DESI1200
 82    4    1     DO IAGRUP=1 TO NAGRUP;    JGL=JNY;                       JNY=LAGRUP(IAGRUP);      DESI1210
 85    4    2     DELTAA=-AREA(JNY)*1E-2;                                                           DESI1220
 86    4    2     AREA(JNY)=AREA(JNY)+DELTAA;                                                       DESI1230
 87    4    2     CALL PROFIL(AREA,INIM,WMOD,LPROF,JNY,JGL);                                        DESI1240
 88    4    2     CALL ANALYS(ELEMTOPO,ELEMGEO,X2,MILIEU,C,STIVHED,KK,KE,F,RESPONS,                 DESI1250
                           EV,T,U);                                                                 DESI1260
                           /* LOOP OVER CRITICAL CONSTRAINTS                                        */DESI1270
 89    4    2     DO IAREST=1 TO NAREST;    IAR=IAREST;                                             DESI1280
 91    4    3     CALL EVALFI(ELEMTOPO,ELEMGEO,MILIEU,STIVHED,RESPONS,STYRKE,FILISTER);             DESI1290
 92    4    3     AK(1+IAREST,1+IAGRUP)=(FIS-FI(IAREST))/DELTAA;  END;                              DESI1300
 94    4    2     AREA(JNY)=AREA(JNY)-DELTAA;                                                       DESI1310
 95    4    2     END;       /* END OF GRADIENT CALCULATION                                         */DESI1320
                           /* SETTING UP THE AUXILLARY LP PROBLEM IN STANDARD FORM                  */DESI1330
 96    4    1     DO IAREST=1 TO NAREST;    AK(1+IAREST,1)=1E0-FI(IAREST);          END;           DESI1340
 99    4    1     DO IAGRUP=1 TO NAGRUP;    IGRUP=LAGRUP(IAGRUP);                                   DESI1350
101    4    2     AK(1,1+IAGRUP)=GRAD(IGRUP);                                                       DESI1360
102    4    2     DO IAREST=1 TO NAREST;                                                            DESI1370
103    4    3     AK(1+IAREST,1)=AK(1+IAREST,1)-AK(1+IAREST,1+IAGRUP)*BL(IGRUP);          END;    DESI1380
105    4    2     AK(1+NAREST+IAGRUP,1)=BU(IGRUP)-BL(IGRUP);                                        DESI1390
106    4    2     AK(1+NAREST+IAGRUP,1+IAGRUP)=1E0;                            END;                 DESI1400
                           /* SOLVING THE STANDARD  LP PROBLEM BY A DUAL STRATEGY                   */DESI1410
108    4    1     NR=1+NAREST+NAGRUP;       NC=1+NAGRUP;                                            DESI1420
110    4    1     CALL DUALP(TABLEAU);                                                              DESI1430
111    4    1     IF TOM=1 THEN             /* NO SOLUTION,PRINT MESSAGE & INCREASE AREA            */DESI1440
112    4    1     DO;                       PUT SKIP DATA(TOM);            DA=BU;           END;    DESI1450
113    4    1             ELSE              /* RETRIEVE SOLUTION OF ORIGINAL LP PROBLEM             */DESI1460
116    4    1     DO;                       DA=0E0;                                                 DESI1470
118    4    2     DO IR=1 TO NR;                                                                    DESI1480
119    4    3     IF IBASIS(IR)<=NAGRUP THEN DA(LAGRUP(IBASIS(IR)))=AK(IR,1);             END;    DESI1490
122    4    2     DA=DA+BL;                 END;                                                    DESI1500
124    4    1     END;          /* END OF REDESIGN LOOP                                             */DESI1510
125    4          ITERAEND:     /* EVALUATION AND PRINTING OF ALL STRUCTURAL CONSTRAINTS            */DESI1520
126    4          MNUD=1;                   REANAL=0;                                               DESI1530
127    4          CALL EVALFI(ELEMTOPO,ELEMGEO,MILIEU,STIVHED,RESPONS,STYRKE,FILISTER);             DESI1540
128    4          END;                                                                              DESI1550
129    3          CALL SKRIV7(STYRKE,STIVHED);                                                      DESI1560
                           /* PRINT ALL NODAL DISPLACEMENTS                                         */DESI1570
130    3          DO IDEL=1 TO NBEL;        CALL SKRIVX(X(IDEL,*),IDEL);           END;             DESI1580
133    3          NSNIT=30;                                                                         DESI1590
134    3          IF FORSTØR>0 THEN         /* PLOT DISPLACED STRUCTURE ON CALCOMP 563              */DESI1600
135    3          CALL PLOTPRU(MAL,ELEMTOPO,ELEMGEO,X1,X2,FORSTØR,MILIEU,EELASTNING,                DESI1610
                           RESPONS,STIVHED);                                                        DESI1620
136    3          IF MAL>0 THEN CALL PLTEND;                                                        DESI1630
138    3          END DESIGN;                                                                       DESI1640
```

```
STMT LEVEL NEST
  1              DUALP:      /* SOLUTION OF A MINIMUM LP PROBLEM BY A DUAL STRATEGY     */DUALP010
                             /* THE PROBLEM IS ASSUMED DUALLY FEASIBLE,I.E. THE COEFFI-*/DUALP020
                             /* CIENTS OF THE MERIT MUST BE POSITIVE                   */DUALP030
                             /* PROBLEM:  MINIMIZE MERIT PROVIDED VECTOR OF CONSTANTS  */DUALP040
                             /* IS >= MATR. OF COEFF. OF RESTRICTS.*VECTOR OF VAR.      */DUALP050
                             /* ON INPUT:                                              */DUALP060
                             /*  A(1,2:NC)      = COEFFICIENTS OF MERIT FUNCTION        */DUALP070
                             /*  A(2:NR,1)      = VECTOR OF CONSTANTS                   */DUALP080
                             /*  A(2:NR,2:NC)=  MATRIX OF COEFFICIENTS OF RESTRICTIONS  */DUALP090
                             /* ON OUTPUT:                                             */DUALP100
                             /* -A(1,1)         = THE MINIMUM MERIT IF TOM=0            */DUALP110
                             /*  A(I,1)         = VALUE OF VARIABLE IB(I)               */DUALP120
                             /*  TOM            = 1 IF NO SOLUTION                      */DUALP130
  2    1         PROCEDURE(TABLEAU);                                                      DUALP140
                 DECLARE (IC,ICP,IR,IRP,ISAVE,ITERA) BIN FIXED,                           DUALP150
                         (CMAX,PIV,RMIN,SAVE) BIN FLOAT(53),                              DUALP160
                       1 TABLEAU,                                                         DUALP170
                         2 (NC,NR,TOM,(IB,NB)(*)) BIN FIXED,                              DUALP180
                         2 A(*,*) BIN FLOAT;                                              DUALP190
                        /* INITIALIZE POINTER ARRAYS IB & NB                           */DUALP200
  3    1         DO IC=1 TO NC;           NB(IC)=IC-1;                      END;          DUALP210
  6    1         IB(1)=NC+NR-1;                                                           DUALP220
  7    1         DO IR=2 TO NR;           IB(IR)=NC+IR-2;                   END;          DUALP230
 10    1         DO ITERA=1 TO 2*NC;                                                      DUALP240
                        /* FIND VARIABLE TO LEAVE BASIS                                */DUALP250
 11    1    1    RMIN=0E0;                                                                DUALP260
 12    1    1    DO IR=2 TO NR;                                                           DUALP270
 13    1    2    IF A(IR,1)<RMIN THEN                                                     DUALP280
 14    1    2    DO;                      RMIN=A(IR,1);       IRP=IR;            END;     DUALP290
 18    1    2    END;                                                                     DUALP300
 19    1    1    IF RMIN=0E0 THEN         RETURN;             /* MINIMUM REACHED       */DUALP310
                        /* FIND VARIABLE TO ENTER BASIS                                */DUALP320
 21    1    1    TOM=1;                   CMAX=-1E16;                                     DUALP330
 23    1    1    DO IC=2 TO NC;                                                           DUALP340
 24    1    2    IF A(IRP,IC)<0E0 THEN                                                    DUALP350
 25    1    2    IF A(1,IC)/A(IRP,IC) > CMAX THEN                                         DUALP360
 26    1    2    DO;                      CMAX=A(1,IC)/A(IRP,IC);  ICP=IC;    TOM=0;END;  DUALP370
 31    1    2    END;                                                                     DUALP380
 32    1    1    IF TOM=1 THEN            RETURN;             /* NO SOLUTION           */DUALP390
                        /* CHANGE OF BASIS                                             */DUALP400
 34    1    1    PIV=A(IRP,ICP);          A(IRP,ICP)=-1E0;                                DUALP410
 36    1    1    DO IR=1 TO NR;           A(IR,ICP)=-A(IR,ICP)/PIV;             END;      DUALP420
 39    1    1    DO IC=1 TO ICP-1,ICP+1 TO NC;                                            DUALP430
 40    1    2    SAVE=A(IRP,IC);          A(IRP,IC)=0E0;                                  DUALP440
 42    1    2    DO IR=1 TO NR;           A(IR,IC)=A(IR,IC)+SAVE*A(IR,ICP);      END;     DUALP450
 45    1    2    END;                                                                     DUALP460
                        /* CHANGE OF BASE POINTERS                                     */DUALP470
 46    1    1    ISAVE=IB(IRP);           IB(IRP)=NB(ICP);    NB(ICP)=ISAVE;               DUALP480
 49    1    1    END;                                                                     DUALP490
 50    1         END DUALP;                                                               DUALP500
```

```
EQUBE:      /*  EQUIVALENT EARTHQUAKE LOADING IS ADDED TO LOADCASE 2  */EQUBE010

STMT LEVEL NEST
  1               EQUBE:     /*  EQUIVALENT EARTHQUAKE LOADING IS ADDED TO LOADCASE 2  */EQUBE010
                  PROCEDURE(UST,X2,WMASSE,W,C,T,BNOEFF);                                 EQUBE020
  2     1         DECLARE ((NNO,DOFNO) EXTERNAL,I,J,IDIS) BIN FIXED,                     EQUBE030
                        (T,V,W,C,WH,(UST,X2,WMASSE,BNOEFF)(*)) BIN FLOAT(53);            EQUBE040
  3     1         W,WH=0E0;                                                              EQUBE050
  4     1         DO I=1 TO NNO;          WH=WH+WMASSE(I)*X2(I);  IDIS=DOFNO*(I-1)+1;    EQUBE060
 .7     1    1    IF UST(IDIS)>0E0 THEN   W=W+WMASSE(I);                                 EQUBE070
  9     1    1    END;                                                                   EQUBE080
 10     1         V=.05E0*C*W*9.81E0/T**.3333333;                                        EQUBE090
 11     1         DO J=1 TO NNO;          I=1+DOFNO*(J-1);                               EQUBE100
 13     1    1    BNOEFF(I)=BNOEFF(I)-V*WMASSE(J)*X2(J)/WH;                              EQUBE110
 14     1    1    END EQUBE;                                                             EQUBE120
```

```
STMT LEVEL NEST
 1            EVALFI:     /* EVALUATION CF STRUCTURAL CCNSTRAINTS
 2     1      PROCEDURE(ELEMTOPC,ELEMGEC,MILIEU,STIVHED,RESPONS,STYRKE,FILISTER);      */EVALFI1C
              DECLARE ((MNUD,FIUD,NELM,NBEL,DOFNO,NGRUP,NCDIS,NREST,                  EVALFI2C
                      REANAL,NAREST,NKRAFT) EXTERNAL,                                  EVALFI3C
                      IR1,IR2,IE1,IE2,IC1,IC2,IB1,IB2,                                EVALFI4C
                      IA,IC,IB,IPRCF,NSNIT,IKE,IR,                                     EVALFI5C
                      I,IELM,IBEL,IGRUP,IENS,ICDIS) BIN FIXED,                         EVALFI6C
                      (RESTEPS EXTERNAL,MG,NCC,NE,ME,L,VF,DVERT,DHORZ,KSI1,KSI2,       EVALFI7C
                      AREAL,COSFI,SINFI, KL,C0,C1,C2,PD,P,KSI,MY,                      EVALFI8C
                      VA,VB,A1,A2,CNE INIT(1),AKL,LI,L1,L1G) BIN FLCAT(53),            EVALFI9C
                      1 ELEMTOPC,                                                      EVALF1CC
                        2 ((LGRUP,LENS,LKE,LNSNIT,LPROF)(*),TOPC(*,*)) EIN FIXED,      EVALF11C
                      1 ELEMGEC,         2 (SL(*),A(*,*),VNF(*)) BIN(53),              EVALF12C
                      1 MILIEU(*),                                                     EVALF13C
                        2 BELASTNING,3 (PJ,BNC,GAMMA)(*) BIN(53),                      EVALF14C
                        2 FLYTNING,  3 (LNCDIS(*) FIXED,UST(*) FLOAT(53)) BIN,         EVALF15C
                      1 STIVHED,         2 ((AREA,IMOM)(*),EMOD) BIN(53),              EVALF16C
                      1 RESPONS(*),      2 (X,KLA,DGA)(*) BIN(53),                     EVALF17C
                      1 STYRKE,          2 (WMOM(*),(SIGMAN,SIGMAB,LIG)(*,*)) BIN(53), EVALF18C
                                         2 (FL,FC,FP,EULERN) BIN FLCAT(53),            EVALF19C
                      1 FILISTER,                                                      EVALF2CC
                        2 (IAR,(LCCIS,LELM,LBEL)(*)) BIN FIXED,                        EVALF21C
 3     1                2 (FIS,(FI,LKSI)(*)) BIN FLOAT(53);                            EVALF22C
 4     1      IF REANAL=1 THEN                                                         EVALF23C
 6     1    1 DO;                IR1,IR2,IR=IAR;                                       EVALF24C
 8     1    1 IB1,IB2=LBEL(IR);  IC1,IC2=LCDIS(IR);                                    EVALF25C
10     1    1 IF IC1>0 THEN GOTC NEDBØJ;                                               EVALF26C
12     1    1 IE1,IE2=LELM(IR);  KSI1,KSI2=LKSI(IR);                                   EVALF27C
13     1    1 GOTO ELEMENTLOOP;                                                        EVALF28C
14     1    1 END;                                                                     EVALF29C
17     1      IR1,IC1,IB1,IE1=1;   KSI1=0E0;                     KSI2=1E0;             EVALF30C
20     1      IE2=NELM;            IB2=NBEL;                     IC2=NCDIS;            EVALF31C
21     1      IR=NGRUP+1;                                                              EVALF32C
24     1      DO I=1 TC NGRUP;     FI(I)=0E0;                    END;                  EVALF33C
27     1      DO I=IR TO NREST;    FI(I)=1E0-RESTEPS;            END;                  EVALF34C
29     1      IF MNUD=1 | FIUD=1 THEN SIGNAL ENDPAGE(SYSPRINT);                        EVALF35C
                         /* STRESS CONSTRAINTS                                         EVALF36C
              ELEMENTLOOP:                                                             */EVALF37C
30     1    1 DO IELM=IE1 TO IE2;                                                      EVALF38C
32     1    1 IGRUP=LGRUP(IELM);   IENS=LENS(IELM);                                    EVALF39C
35     1    1 IKE=LKE(IELM);       NSNIT=LNSNIT(IELM);          IPRCF=LPROF(IGRUP);    EVALF40C
37     1    1 AREAL=AREA(IGRUP);   L=SL(IENS);                                         EVALF41C
39     1    1 COSFI=A(IENS,1);     SINFI=A(IENS,2);                                    EVALF42C
41     1    1 IA=DOFNO*TOPO(IELM,1); IB=DOFNO*TOPO(IELM,2);                            EVALF43C
43     1    1 P    =AREAL*L*COSFI; MY=EMCD*IMCM(IGRUP)/L;                              EVALF44C
44     1    1 LI=L*SQRT(AREAL/IMCM(IGRUP));                                            EVALF45C
45     1    1 IF MNUD=1 THEN                                                           EVALF46C
              PUT EDIT('ELEM BEL       KSI      FI     N.FORCE     MOMENT      KL       LI',EVALF47C
                       '      DG      VNF','  ')                                       EVALF48C
46     1    1          (SKIP(2),A,A,SKIP,A);                                           EVALF49C
47     1    2 DO IBEL=IB1 TO IB2;     /* LOCP CVER LOADCASES                           EVALF5CC
49     1    2 KL=KLA(IBEL,IELM);      AKL=ABS(KL);                                     */EVALF51C
              PD=(P*GAMMA(IBEL,IPRCF)+PJ(IBEL,IELM))*L/MY;      /* CIMENSICNLESS LCAD  EVALF52C
                         /* ELEMENT DEFORMATION PARAMETERS                             */EVALF53C
50     1    2 DHORZ=X(IBEL,IB-2)-X(IBEL,IA-2);                                         */EVALF54C
51     1    2 DVERT=X(IBEL,IB-1)-X(IBEL,IA-1);                                         EVALF55C
                                                                                       EVALF56C
```

```
STMT LEVEL NEST

 52    1    2   VF=(DHORZ*SINFI-DVERT*COSFI)/L;                                              EVALF570
 53    1    2   VA=-VF-X(IBEL,IA);        VB= VF+X(IBEL,IB);                                 EVALF580
 55    1    2   IF NKRAFT=1 & AKL>.15 THEN                                                   EVALF590
 56    1    2   CALL KLVIRKN(KL,A1,A2,'EVALFI',LI,VA,VB,PD,IKE,C0,C1,C2,L,L,L,L);            EVALF600
 57    1    2                                ELSE                                            EVALF610
 57    1    2   DO;                                                                          EVALF620
 58    1    3   IF IKE=4 THEN                                                                EVALF630
 59    1    3   DO;       /* IN COMPRESSION ECCENTRICITY & INIT.CURVATURE IB INCLUD.*/EVALF640
 60    1    4   IF KL>0E0 THEN                                                               EVALF650
 61    1    4   IF PD>=0E0 THEN C0=1E0; ELSE C0=-1E0;              ELSE C0=CEC;              EVALF660
 65    1    4   PD=PD+C0*16E-3*KL2;                                                          EVALF670
 66    1    4   VA,VB=PD/24E0+C0*KL*AKL/(40E0*LI);                                           EVALF680
 67    1    4   END;                                                                         EVALF690
 68    1    3   C0=4E0*VA-2E0*VB-PD/12E0;     C2=-PD/2EC;          C1=GE0*(VB-VA)-C2;         EVALF700
 71    1    3   END;                                                                         EVALF710
 72    1    2   NE=-AKL*KL*NY/L;                        /* NORMALFORCE,CONSTANT             */EVALF720
 73    1    2   NOO=AREAL*SIGMAN(IBEL,IPROF);                         /* ALLOW. TENSION     */EVALF730
 74    1    2   IF KL>0 THEN                                          /* ALLOW. COMPRESSICN */EVALF740
 75    1    2   DO;                     LILIG=LI/LIG(IBEL,IPROF);                            EVALF750
 77    1    3   IF LILIG>1E0 THEN                                     /* EULER DOMAIN       */EVALF760
 78    1    3   NOO=NCO*FP/LILIG**2;                                                         EVALF770
 79    1    3             ELSE                                        /* PLASTIC DOMAIN     */EVALF780
 79    1    3   NOO=NOO*(FC-FL*FP*LILIG**2);                                                 EVALF790
 80    1    3   END;                                                                         EVALF800
 81    1    2   MO =WMOM(IGRUP)*SIGMAE(IBEL,IPROF);/* ALLOW. MOMENT                         EVALF810
 82    1    2   DO KSI=KSI1 TO KSI2 BY ONE/(NSNIT-1);                 /* SECTION LOOP       */EVALF820
 83    1    3   IF NKRAFT=1 & AKL>.15 THEN              /* NONLIN. GEOM. MOMENT             */EVALF830
 84    1    3   IF KL>0 THEN                                          /* COMPRESSION        */EVALF840
 85    1    3   ME=MY*(-C0 +C1*AKL*SIN (KL*KSI) + C2*AKL*KL*COS (KL*KSI));                   EVALF850
 86    1    3           ELSE                                          /* TENSION            */EVALF860
 87    1    3   ME=MY*(-C0 +C1*AKL*SINH(KL*KSI) + C2*AKL*KL*COSH(KL*KSI));                   EVALF870
 87    1    3                                ELSE    /* LINEAR GEOM. MOMENT                 */EVALF880
 88    1    3   ME=MY*(C0+KSI*(C1+KSI*C2));                                                  EVALF890
 89    1    3   FIS=ABS(NE)/NOO+ABS(ME)/MO ;                                                 EVALF900
 89    1    3   IF MNUD=1 THEN                                                               EVALF910
 90    1    3   PUT EDIT(IELM,IBEL,KSI,FIS,NE,ME,KL,LI,DGA(IBEL,IELM),VNF(IELM))              EVALF920
                      (SKIP,2 F(4),2 F(8,4),2 F(10,3),F(8,4),F(8,2),2 E(9,2));               EVALF930
 91    1    3   IF REANAL=1 THEN GOTO UDFI;                                                  EVALF940
                          /* CRITICAL VALUES OF FI ARE STORED                              */EVALF950
 93    1    3   IF FIS>FI(IGRUP) THEN DO;                          FI(IGRUP)=FIS;            EVALF960
 96    1    4   LKSI(IGRUP)=KSI;          LELM(IGRUP)=IELM;        LBEL(IGRUP)=IBEL; END;    EVALF970
100    1    3                   ELSE                                                         EVALF980
100    1    3   IF FIS>FI(IR) THEN DO;                             FI(IR)=FIS;               EVALF990
103    1    4   LKSI(IR)=KSI;             LELM(IR)=IELM;           LBEL(IR)=IBEL;            EVAL1000
106    1    4   DO I=NGRUP+1 TO NREST; IF FIS>FI(I) THEN DO;       FIS=FI(I);   IR=I;        EVAL1010
111    1    6   END; END;                                                                    EVAL1020
113    1    4   END ELEMENTLOOP;                                                             EVAL1030
                        /* DISPLACEMENT CONSTRAINTS                                        */EVAL1040
117    1        LCDIS=0 ;                                                                    EVAL1050
118    1        NEDBØJ:                                                                      EVAL1060
119    1    1   DO IBEL=IB1 TO IB2;      DO IC=IC1 TO IC2;         ICDIS=LNCDIS(IBEL,IC);    EVAL1070
121    1    2   IF ICDIS=0 THEN FIS=0; ELSE                                                  EVAL1080
123    1    2   FIS=ABS(X(IBEL,ICDIS))/UST(IBEL,ICDIS);                                      EVAL1090
124    1    2   IF MNUD=1 THEN                                                               EVAL1100
125    1    2   PUT EDIT('      BEL            FI      ICDIS            X')(SKIP(2),A)       EVAL1110
```

```
STMT LEVEL NEST

                        (IBEL,FIS,ICDIS,X(IBEL,ICDIS))                                  EVAL1120
                        (SKIP,F(8),F(16,4),F(10),E(11,2));                              EVAL1130
 126    1    2    IF REANAL=1 THEN GOTC UDFI;                                           EVAL1140
 128    1    2    IF FIS>FI(IR) THEN DC; LCDIS(IR)=IC;          LBEL(IR)=IBEL;          EVAL1150
 132    1    3    FI(IR)=FIS;                                                           EVAL1160
 133    1    3    DO I=NGRUP+1 TO NREST; IF FIS>FI(I) THEN DO;  FIS=FI(I);              EVAL1170
 137    1    5    IR=I;                                                                 EVAL1180
 138    1    5    END NEDBØJ;                                                           EVAL1190
 143    1         NAREST=NREST;                                                         EVAL1200
 144    1         DO I=NREST TO NGRUP+1 BY-1 WHILE (FI(I)=1E0-RESTEPS);                 EVAL1210
 145    1    1    NAREST=I-1;               END;                                        EVAL1220
 147    1         UDFI:                                                                 EVAL1230
 148    1         IF REANAL=0 THEN IR2=NAREST;                                          EVAL1240
 149    1         IF FIUD=1 THEN DO;                                                    EVAL1250
 151    1    1    PUT EDIT('ELEM BEL      KSI      FI      ICDIS')(SKIP(2),A)            EVAL1260
                        ((LELM(I),LBEL(I),LKSI(I),FI(I),LCDIS(I) DO I=IR1 TO IR2))      EVAL1270
                        (SKIP,(NAREST)(SKIP,2 F(4),2 F(8,4),F(10)));                    EVAL1280
 152    1    1    END EVALFI;                                                           EVAL1290
```

```
KEIKK:       /* 1-DIM. ELEM.STIFFN.MATRIX IS PLACED IN GLOB.STIFFN.MAT.*/KEIKKC10                PAGE

STMT LEVEL NEST
  1                KEIKK:      /* 1-DIM. ELEM.STIFFN.MATRIX IS PLACED IN GLOB.STIFFN.MAT.*/KEIKK010
                   PROCEDURE(KK,KE,TOPC,NOPEL,DCFNC);                                    KEIKK020
  2      1         DECLARE (NON1,NON2,KE1,KK1,KK2,KKB1,KKB2,K,NOPEL,DCFNC,DCFNOH,       KEIKK030
                           TOPO(*)) BIN FIXED,                                            KEIKK040
                           (KE(*),KK(*,*)) BIN FLOAT(53);                                 KEIKK050
  3      1         KE1=0;                 DCFNOH=DCFNC-1;                                 KEIKK060
  5      1         DO NON1=1 TO NOPEL;    KKB1,K=DCFNC*(TOPC(NON1)-1)+1;                  KEIKK070
  7      1    1    DO NON2=NON1 TO NOPEL; KKB2  =DCFNC*(TOPC(NON2)-1)+2-KKE1;             KEIKK080
  9      1    2    IF KKB2>=1 THEN                                                        KEIKK090
 10      1    2    DO KK1=KKB1 TO KKB1+DCFNOH;                                            KEIKK100
 11      1    3    DO KK2=MAX(KKB2,1) TO KKB2+DCFNOH;             KE1=KE1+1;              KEIKK110
 13      1    4    KK(KK1,KK2)=KK(KK1,KK2)+KE(KE1);                                       KEIKK120
 14      1    4    END;                   KKB2=KKB2-1;                                    KEIKK130
 16      1    3    END;                   ELSE                                            KEIKK140
 17      1    2    DO;                    KKE1=KKB1+KKB2-1;               KKE2=K+1-KKB1;  KEIKK150
 20      1    3    DO K=KKB2 TO KKB2+DCFNOH;                              KK2=K;          KEIKK160
 22      1    4    DO KK1=KKB1 TO KKB1 + DCFNOH;                          KE1=KE1+1;      KEIKK170
 24      1    5    KK(KK1,KK2)=KK(KK1,KK2)+KE(KE1);                       KK2=KK2-1;      KEIKK180
 26      1    5    END; END;  END;  END;                                                  KEIKK190
 30      1    1    END KEIKK;                                                             KEIKK200
```

```
STMT LEVEL NEST
  1               KEMAT:      /* ELEM.STIFFN.MATRIX COMPUTED AND STORED 1-DIMENSIONALLY */KEMAT010
                  PROCEDURE(L,A,AREA,IMOM,EMOD,KL,IKE,KE);                                  KEMAT020
  2     1         DECLARE ((NKRAFT,ISTAB)EXTERNAL,IKE) BIN FIXED,                           KEMAT030
                         (EMOD,KL,AKL,AREA,IMOM,MY,A1,A2,M,S,C,L1,L2,L3,L,                  KEMAT040
                         (KE,A)(*)) BIN FLOAT(53);                                          KEMAT050
  3     1         L1=EMOD*AREA/L;                                                           KEMAT060
 .4     1         MY=EMOD*IMOM/L;                                                           KEMAT070
  5     1         AKL=ABS(KL);                                                              KEMAT080
  6     1         IF IKE=4 THEN           A1,A2=0E0;                                        KEMAT090
  8     1                  ELSE                                                             KEMAT100
  8     1         IF IKE=1 THEN                                                             KEMAT110
  9     1         IF NKRAFT=1 & AKL>.15 THEN                                                KEMAT120
 10     1         CALL KLVIRK (KL,A1,A2,'KEMAT');                                           KEMAT130
 11     1                             ELSE                                                  KEMAT140
 11     1         DO;                  A1=4E0;            A2=-2E0;            END;          KEMAT150
 15     1     .   L3=(A1-A2)*MY/L;     L2=2E0*L3/L;                                         KEMAT160
 17     1         IF ISTAB>0 THEN      L2=L2-MY*AKL*KL/L**2;                                KEMAT170
 19     1         C=A(1);              S=A(2);                                              KEMAT180
                        /* SPEC. 1-DIM. STORAGE OF UPP.TRIANG. OF KE IN GLOB.DATUM*/KEMAT190
 21     1         KE( 1),KE(16),M=C**2*L1+S**2*L2;       KE( 7)=-M;                         KEMAT200
 23     1         KE( 2),KE(17),M=C*S*(L1-L2);           KE( 8),KE(10)=-M;                  KEMAT210
 25     1         KE( 3),KE( 9),M=-S*L3;                 KE(13),KE(18)=-M;                  KEMAT220
 27     1         KE( 4),KE(19),M=S**2*L1+C**2*L2;       KE(11)=-M;                         KEMAT230
 29     1         KE( 5),KE(12),M=C*L3;                  KE(14),KE(20)=-M;                  KEMAT240
 31     1         KE( 6),KE(21)  =A1*MY;                 KE(15)=-A2*MY;                     KEMAT250
 33     1         END KEMAT;                                                                KEMAT260
```

```
STMT LEVEL NEST
  1                 KLVIRKN:   /* EFFECTS FROM  NONLINEAR GEOMETRIC BEAM ANALYSIS            */KLVIRK10
                    PROCEDURE(KL,A1,A2,OPT,LI,VA,VB,PD,IKE,C0,C1,C2,C3,VF,L,DG);              KLVIRK20
  2     1           KLVIRK:    /* SHORT PARAMETERLIST,USED WHEN CALLED FROM KEMAT OR BEL    */KLVIRK30
                    ENTRY(KL,A1,A2,OPT);                                                      KLVIRK40
  3     1           DECLARE (KL,KL2,LI,VA,VB,PD,C0,C1,C2,C3,A1,A2,B1,D,CKL,KLSKL,             KLVIRK50
                            VF,L,DG)BIN FLOAT(53),                                            KLVIRK60
                            OPT CHAR(*),                                                      KLVIRK70
                            IKE BIN FIXED(15);                                                KLVIRK80
  4     1           KL2=ABS(KL)*KL;                                                           KLVIRK90
  5     1           IF KL> 0 THEN               /* COMPRESSION         */                     KLVIR100
  6     1           DO;                         KLSKL= KL*SIN (KL);    CKL=COS (KL);    END;  KLVIR110
 10     1                 ELSE                  /* TENSION             */                     KLVIR120
 10     1           DO;                         KLSKL=-KL*SINH(KL);    CKL=COSH(KL);    END;  KLVIR130
 14     1           D=2E0-2E0*CKL-KLSKL;                                                      KLVIR140
 15     1           A1=(KLSKL-KL2*CKL)/D;                                                     KLVIR150
 16     1           A2=(KLSKL-KL2     )/D;                                                    KLVIR160
 17     1           IF OPT='KEMAT' THEN RETURN;                                               KLVIR170
 19     1           IF OPT='BEL' THEN           /* WHEN CALLED A1,A2 YIELD FIXITYCOEFFICIENTS */KLVIR180
 20     1           DO;                         A1,A2=(2E0-(A1+A2))/(2E0*KL2);                KLVIR190
 22     1    1      RETURN;                     END;                                          KLVIR200
 24     1           IF IKE=4 THEN                                                             KLVIR210
 25     1           DO;         /* IN COMPRESSION ECCENTRICITY & INIT.CURVATURE IS INCLUD.*/KLVIR220
 26     1    1      IF KL>0E0 THEN                                                            KLVIR230
 27     1    1      IF PD>=0E0 THEN C0=1E0; ELSE C0=-1E0;         ELSE C0=0E0;                KLVIR240
 31     1    1      PD=PD*C0*16E-3*KL2;                                                       KLVIR250
 32     1    1      VA,VB=PD*D/(2E0*KL2*KLSKL)+C0*KL2/((A1+A2)*20E0*LI);                      KLVIR260
 33     1    1      IF OPT¬='EVALFI' THEN VA,VB=VB+C0*8E-3;                                   KLVIR270
 35     1    1      END;                                                                      KLVIR280
 36     1           A1=A1/KL2;                  A2=A2/KL2;                                    KLVIR290
 38     1           B1=A1-A2;                                                                 KLVIR300
 39     1           C0=PD/KL2;                                                                KLVIR310
 40     1           C1=(1E0-B1)*VA + B1*VB + C0/2E0;                                          KLVIR320
 41     1           C2=      A1*VA + A2*VB + C0/2E0*(A1+A2);                                  KLVIR330
 42     1           IF OPT='EVALFI' THEN RETURN;                                              KLVIR340
 44     1           C3=      B1*(VA-VB)      - C0/2E0 - VF;                                   KLVIR350
 45     1           IF OPT='FLOTPR' THEN RETURN;                                              KLVIR360
 47     1           DG=(C0**2/3E0 + C3**2 + C0*C3 + (C1**2+C2**2*KL2)/2E0                     KLVIR370
                      + CKL*KLSKL*(C1**2/KL2-C2**2)/2E0 - 2E0*D*C0*(C1*(1E0-B1)/KL2+C2*A1) KLVIR380
                      - C1*C2*KLSKL**2/KL2 + 2E0*C3*(C1*KLSKL/KL2+C2*(CKL-1E0)))*L/2E0;     KLVIR390
 48     1           IF IKE=4 & KL>0E0 THEN DG=DG-32E-6*L/3E0;                                 KLVIR400
 50     1           END KLVIRKN;                                                              KLVIR410
```

```
STMT LEVEL NEST
  1                KONVERG:    /* CHECK CONVERGENCE OF KLA & DGA IF ISTAB>0          */KONVER10
                   PROCEDURE(MILIEUB,ELEMTOPC,ELEMGEO,STIVHED,RESPONS,NYIT,IBEL);      KONVER20
  2    1           DECLARE ((NELM,DOFNO,KORDE,NLKRUM,ISTAB,AUTOVNF) EXTERNAL,          KONVER30
                         IBEL,IELM,IKE,IENS,IGRUP,IA,IB,NYIT) BIN FIXED,               KONVER40
                         (COSFI,SINFI,L,DHORZ,DVERT,VF,VA,VB,DN,DG,PD,RAF,             KONVER50
                         A1,A2,C0,C1,C2,C3,KL,AKL,LI) BIN FLOAT(53),                   KONVER60
                         1 ELEMTOPC,                                                   KONVER70
                           2 ((LGRUP,LENS,LKE,LNSNIT,LPROF)(*),TOPC(*,*)) BIN FIXED,   KONVER80
                         1 ELEMGEO,       2 (SL(*),A(*,*),VNF(*),EPSF) BIN(53),        KONVER90
                         1 MILIEUB,       2 (PJ,BNG,GAMMA)(*) BIN FLOAT(53),           KONVE100
                         1 STIVHED,       2 ((AREA,INOM)(*),EMOD) BIN FLOAT(53),       KONVE110
                         1 RESPONS,       2 (X,KLA,DGA)(*) BIN FLOAT(53);              KONVE120
  3    1           RAF=.01;                                                            KONVE130
  4    1           DO IELM=1 TO NELM;      IENS=LENS(IELM);       IGRUP=LGRUP(IELM);   KONVE140
  7    1    1      COSFI=A(IENS,1);        SINFI=A(IENS,2);       L=SL(IENS);          KONVE150
 10    1    1      IA=DOFNO*TOPC(IELM,1); IB=DOFNO*TOPC(IELM,2);                       KONVE160
 12    1    1      IKE=LKE(IELM);                                                      KONVE170
 13    1    1      LI=L*SQRT(AREA(IGRUP)/INOM(IGRUP));                                 KONVE180
 14    1    1      DHORZ=X(IB-2)-X(IA-2); DVERT=X(IB-1)-X(IA-1);                       KONVE190
 16    1    1      DN= DHORZ*COSFI+DVERT*SINFI + DGA(IELM);                            KONVE200
 17    1    1      IF AUTOVNF=1 & IKE=4 & IBEL=1 THEN VNF(IELM)=-DN+L*EPSF;            KONVE210
 19    1    1      DN=DN+VNF(IELM);                                                    KONVE220
 20    1    1      KL=-SIGN(DN)*LI*SQRT(ABS(DN)/L);                  AKL=ABS(KL);      KONVE230
 22    1    1      VF=(DHORZ*SINFI-DVERT*COSFI)/L;                                     KONVE240
 23    1    1      IF KORDE=0 THEN          DG=0E0;                                    KONVE250
 25    1    1                 ELSE                                                     KONVE260
 25    1    1      DO;                                                                 KONVE270
 26    1    2      VA = VF-X(IA);               VB = -VF+X(IB);                        KONVE280
 28    1    2      PD=(PJ(IELM)+GAMMA(LPROF(IGRUP))*AREA(IGRUP)*L*COSFI)               KONVE290
                         *L**2/(EMOD*INOM(IGRUP));                                     KONVE300
 29    1    2      IF NLKRUM=1 & AKL>.1E THEN                                          KONVE310
 30    1    2      CALL KLVIRKN(KL,A1,A2,'KONVERG',LI,VA,VB,PD,IKE,                    KONVE320
                              C0,C1,C2,C3,VF,L,DG);                                    KONVE330
 31    1    2                               ELSE                                       KONVE340
 31    1    2      DO;                                                                 KONVE350
 32    1    3      IF IKE=4 THEN                                                       KONVE360
 33    1    3      DO;       /* IN COMPRESSION ECCENTRICITY & INIT.CURVATURE IS INCLUD.*/KONVE370
 34    1    4      IF KL>0E0 THEN                                                      KONVE380
 35    1    4      IF PD>=0E0 THEN C0=1E0; ELSE C0=-1E0;           ELSE C0=0E0;        KONVE390
 39    1    4      PD=PD+C0*16E-3*KL*AKL;                                              KONVE400
 40    1    4      VA,VB=PD/24E0+C0*KL*AKL/(40E0*LI)+C0*8E-3;                          KONVE410
 41    1    4      END;                                                                KONVE420
 42    1    3      DG=    L*(VF**2/2E0+(VA**2+VB**2)/15E0+VA*VB/30E0)                  KONVE430
                         +PD*(PD+(VA+VB)*84E0)*L/60480E0;                              KONVE440
 43    1    3      IF IKE=4 & KL>0E0 THEN DG=DG-32E-6*L/3E0;                           KONVE450
 45    1    3      END; END;                                                           KONVE460
 47    1    1      IF ISTAB>0 THEN                                                     KONVE470
 48    1    1      IF (ABS(KL-KLA(IELM))> RAF*ABS(KLA(IELM))& ABS(KL)>1E-3) |          KONVE480
 49    1    1        (ABS(DG-DGA(IELM))> RAF*ABS(DGA(IELM)) & ABS(DG)>1E-15) THEN NYIT=1; KONVE490
 50    1    1      KLA(IELM)=KL;           DGA(IELM)=DG;                 END;          KONVE500
 53    1           END KONVERG;                                                        KONVE510
                                                                                       KONVE520
```

```
LÆS:          /* INPUT:DIRECT AND/OR GENERATED                                         */LÆS00010

STMT LEVEL NEST
  1                    LÆS:       /* INPUT:DIRECT AND/OR GENERATED                                  */LÆSC0010
                       PROCEDURE;                                                                      LÆSC0020
  2      1             DECLARE ((ANSVING,AUTOPO,AUTOBE,AUTOGE,NNO,NELM,DOFNO,DOFEL,NOPEL,              LÆSC0030
                               NITERA,NFSI,NNUD,FIUD,NEV,NAGRUP,AUTOVNF,                                LÆSC0040
                               NDIS,NVERT,NHORZ,ISTAB,KORDE,NKRAFT,NLKRUM,MAL,FORSTØR,                  LÆSC0050
                               NBEL,NBAND,NGRUP,NCDIS,NREST,NPROF,NENS)EXTERNAL,                        LÆSC0060
                               I,J,K,L,IELM,IENS,JELM,NON1,IA,IB,NSNIT,IDIS, IKE,                       LÆSC0070
                               IG,IE,JA,JB,IEV,NSPEC,JNOD,                                              LÆSC0080
                               LAGRUP(*),LNSNIT(NELM),LNCDIS(NBEL,NCDIS))BIN FIXED,                     LÆSC0090
                               HOVED CHAR(100) EXTERNAL,                                                LÆSC0100
                               ((LHORZ,LVERT,RESTEPS,DELTAA,RELAXF,RELAXR,HORZEEL,                      LÆSC0110
                               REDUKF,REDUKR,FFM,ALPHA,BETA) EXTERNAL,                                  LÆSC0120
                               ENOD,ENODA,SIGMN,SIGMB,GAMMAP,C,CA,PIH,KSI,ONE INIT(1),                  LÆSC0130
                               LIMAX,FAC,FC,FP,EULERN,(GRAD,X1,X2,FM,DA,AMIN,AMAX)(*),                  LÆSC0140
                               (BND,UST)(NBEL,NDIS),(GAMMA,SIGMAN,SIGMAB)(NBEL,NPROF),                  LÆSC0150
                               EV(*,*),PJ(NBEL,NELM)) BIN FLOAT(53),                                    LÆSC0160
                               1 MILIEU(*),                                                             LÆSC0170
                                 2 BELASTNING,3 (PJA,BNDA,GAMMAA)(*) BIN FLOAT(53),                     LÆSC0180
                                 2 FLYTNING,  3 (LNCDISA(*) FIXED,USTA(*) FLOAT(53)) BIN,               LÆSC0190
                               1 STYRKE,      2 (MMOM(*),SIGMANA,SIGMABA,LIG)(*,*)) BIN(53),            LÆSC0200
                                              2 (FL,FCA,FPA,EULERNA) BIN FLOAT(53),                     LÆSC0210
                               1 ELEMTOPO,                                                              LÆSC0220
                                 2 ((LGRUP,LENS,LKE,LNSNITA,LPROF)(*),TOPO(*,*)) BIN FIXED,             LÆSC0230
                               1 ELEMGEO,     2 (SL(*),A(*,*),VNF(*),EPSF) BIN(53);                     LÆSC0240
                                 /* DESCISION & DIMENSIONING VARIABLES,HEADING,NODAL DATA           */LÆSC0250
  3      1             GET EDIT(HOVED)(A(80));                                                         LÆSC0260
  4      1             SUBSTR(HOVED,80,21)='DATE:'||DATE||' TIME:'||SUBSTR(TIME,1,4);                  LÆSC0270
  5      1             GET DATA;                                                                       LÆSC0280
  6      1             IF AUTOGE=1 THEN DO:      /* GENERATION OF NODAL DATA OF STANDARD FRAME */LÆSC0290
  8      1    1        LVERT=LVERT/NVERT;        LHORZ=LHORZ/NHORZ;        NNC=(NVERT+1)*(NHORZ+1);LÆSC0300
 11      1    1        NELM=NVERT*(2*NHORZ+1);END;                                                     LÆSC0310
 13      1             DOFEL=DOFNO*NOPEL;        NDIS=DOFNO*NNO;                                       LÆSC0320
 15      1             NREST=NGRUP+NBEL*(NGRUP+NCDIS);                                                 LÆSC0330
 16      1             IF ISTAB<2|ISTAB=3 THEN KORDE=0;ELSE KORDE=1;                                   LÆSC0340
 19      1             IF ISTAB<3  THEN NKRAFT=0;ELSE NKRAFT=1;                                        LÆSC0350
 22      1             IF ISTAB=5 THEN NLKRUM=1; ELSE NLKRUM=0;                                        LÆSC0360
 25      1             RETURN ;                                                                        LÆSC0370
 26      1             LÆS2:      /* TOPOLOGY,COORDINATES,BANDWIDTH                                 */LÆSC0380
                       ENTRY(ELEMTOPO,X1,X2);                                                          LÆSC0390
 27      1             IF AUTOGE=1 THEN                                                                LÆSC0400
 28      1             DO I=0 TO NVERT;          DO J=0 TO NHORZ;          NON1=1+J+I*(NHORZ+1);   LÆSC0410
 31      1    2        X1(NON1)=LHORZ*J ;        X2(NON1)=LVERT*I;         END;   END;             LÆSC0420
 35      1                        ELSE                                                                 LÆSC0430
 35      1             GET LIST((K,X1(K),X2(K) DO I=1 TO NNO));                                        LÆSC0440
 36      1             IF AUTOPO=1 THEN DO;      IELM=NNC-NHORZ-1;                                     LÆSC0450
 39      1    1        DO NON1=1 TO IELM;                                                              LÆSC0460
 40      1    2        TOPO(NON1,1)=NON1;        TOPO(NON1,2)=NON1+NHORZ+1;      END;                  LÆSC0470
 43      1    1        DO J=0 TO NHORZ-1;        DO I=1 TO NVERT;                                      LÆSC0480
 45      1    3        IELM=IELM+1;              NON1=1+J+I*(NHORZ+1);                                 LÆSC0490
 47      1    3        TOPO(IELM,1)=NON1;        TOPO(IELM,2)=NON1+1 ;           END;  END:  END;      LÆSC0500
 52      1                     ELSE                                                                    LÆSC0510
 52      1             GET LIST((K,TOPO(K,1),TOPO(K,2) DO I=1 TO NELM));                               LÆSC0520
                                 /* GROUPING AND END FIXITY                                         */LÆSC0530
 53      1             GET LIST(LGRUP,LKE);                                                            LÆSC0540
                                 /* GEOMETRICALLY ALIKE ELEMENTS                                    */LÆSC0550
 54      1             DO IELM=1 TO NELM;                                                              LÆSC0560
```

```
LÆS.        /* INPUT:DIRECT AND/CR GENERATED                                            */LÆS00010

STMT LEVEL NEST

 55    1    1   IA=TOPO(IELM,1);          IB=TOPO(IELM,2);                                        LÆSC0570
 57    1    1   DO JELM=IELM-1 BY -1 TO 1;                                                         LÆSC0580
 58    1    2   JA=TOPO(JELM,1);          JB=TOPO(JELM,2);                                        LÆSC0590
 60    1    2   IF ABS(X2(IA)-X2(IB)-X2(JA)+X2(JB))<1E-3  &                                         LÆSC0600
                   ABS(X1(IA)-X1(IB)-X1(JA)+X1(JB))<1E-3  THEN                                     LÆSC0610
 61    1    2   DO;                       LENS(IELM)=LENS(JELM);  GO TO ENSEND;      END;        LÆSC0620
 65    1    2   END;                                                                               LÆSC0630
 66    1    1   NENS=NENS+1;              LENS(IELM)=NENS;                                        LÆSC0640
 68    1    1   ENSEND:END;                                                                        LÆSC0650
                      /* BANDWICTH:NBAND                                                       */LÆSC0660
 69    1        DO I=1 TO NELM;           NBAND=MAX(NBAND,TOPO(I,2)-TOPO(I,1));      END;        LÆSC0670
 72    1        NBAND=(NBAND+1)*DOFNC-1  ;                                                         LÆSC0680
 73    1        RETURN ;                                                                           LÆSC0690
 74    1        LÆS3:         /* MILIEU- & STRENGTH ,GEOMETRY,FABRIC. BOUNDS ON AREAS           */LÆSC0700
                      /* GRAD. CF MERIT,ACTIVE VAR.,INIT. OF EIGENVECTORS                      */LÆSC0710
                ENTRY(ELEMTOPO,ELEMGEO,X1,X2,MILIEU,CA,STYRKE,EMODA,EV,                            LÆSC0720
                      DA,AMIN,AMAX,FP,LAGRUP,GRAD);                                                LÆSC0730
                      /* COSFI & SINFI                                                         */LÆSC0740
 75    1        DO IELM=1 TO NELM;        IENS=LENS(IELM);                                        LÆSC0750
 77    1    1   IF SL(IENS)=0E0 THEN DO;                                                           LÆSC0760
 79    1    2   IA=TOPO(IELM,1);          IB=TOPO(IELM,2);                                        LÆSC0770
 81    1    2   SL(IENS)=SQRT((X1(IB)-X1(IA))**2+(X2(IB)-X2(IA))**2);                              LÆSC0780
 82    1    2   A(IENS,1)=(X1(IB)-X1(IA))/SL(IENS);          /* COSFI   */                        LÆSC0790
 83    1    2   A(IENS,2)=(X2(IB)-X2(IA))/SL(IENS);          /* SINFI   */                        LÆSC0800
 84    1    2   END; END;                                                                          LÆSC0810
 86    1        FC=8E-1;                  FP=4E-1;                  EULERN=17E-1;                 LÆSC0820
                      /* SOME STANDARD VALUES OF POSSIBLE INPUT                                */LÆSC0830
 89    1        GET DATA(NSNIT,GAMM,SIGMN,SIGMB,FC,FP,EULERN);                                     LÆSC0840
 90    1        FL=FC/FP-1E0;             PIH=4E0*ATAN(CNE);                                      LÆSC0850
 92    1        LNSNIT=NSNIT;             GAMMA=GAMM;                                             LÆSC0860
 94    1        SIGMAN=SIGMN;             SIGMAB=SIGMB;                                           LÆSC0870
 96    1        BNO=0E0;                  PJ=0E0;                                                 LÆSC0880
 98    1        UST=1E20;                 LNCDIS=0;                 VNF=0E0;                      LÆSC0890
                      /* IF INTERNAL PIN CONN. THEN UST=0  => KK(IDIS,1)=1E20              */LÆSC0900
101    1        DO IDIS=3 BY 3 TO NDIS;UST(*,IDIS)=0E0;                 END;                     LÆSC0910
104    1        DO IELM=1 TO NELM;        IKE=LKE(IELM);                                          LÆSC0920
106    1    1   IF IKE=1 THEN UST(*,3*TOPC(IELM,1)),UST(*,3*TOPO(IELM,2))=1E20;                    LÆSC0930
108    1    1            ELSE                                                                      LÆSC0940
108    1    1   IF IKE<4 THEN UST(*,3*TOPC(IELM,4-IKE))=1E20;                                      LÆSC0950
110    1    1   END;                                                                               LÆSC0960
111    1        IF AUTOGE=1  THEN         DO I=1 TO DOFNO*(NHORZ+1);                              LÆSC0970
113    1    1   UST(*,I)=0E0;             END;                                                    LÆSC0980
115    1        GET DATA(LNSNIT,GAMMA,SIGMAN,SIGMAB,EMOD,BND,PJ,C,UST,LNCDIS);                     LÆSC0990
116    1        IF ALTOVNF=2 THEN         GET LIST(VNF);                                          LÆSC0995
118    1        IF AUTOBE=1  THEN         /* ELEM LOADS BY SPECIAL LIST FORM                   */LÆSC1000
119    1        DO;                       GET DATA(NSPEC);                                        LÆSC1010
121    1    1   DO I=1 TO NSPEC;          /* JA IB FIRST,JA IB LAST ELEMENTNUMBER              */LÆSC1020
122    1    2   GET DATA(IBEL,JA,JNCR,JB,P);                                                       LÆSC1030
123    1    2   DO J=JA BY JNCR TO JB;    PJ(IBEL,J)=P;             END;  END;  END;              LÆSC1040
128    1        DO I=1 TO NELM;           IE=LENS(I);                                             LÆSC1050
130    1    1   DO J=1 TO NBEL;           PJ(J,I)=PJ(J,I)*SL(IE);  END;  END;                     LÆSC1060
134    1        SIGMANA=SIGMAN;           SIGMABA=SIGMAB;           LNSNITA=LNSNIT;               LÆSC1070
137    1        GAMMAA=GAMMA;             CA=C;                     USTA=UST;                     LÆSC1080
140    1        BNOA=BNO;                 PJA=PJ;                   LNCDISA=LNCDIS;               LÆSC1090
143    1        EMODA=EMOD;                                                                        LÆSC1100
```

```
LÆS:        /* INPUT:DIRECT AND/CR GENERATED                                    */LÆS00010

STMT LEVEL NEST

144   1          FCA=FC;                  FPA=FP;                  EULERNA=EULERN;              LÆSC1110
147   1          LIG=PIH*SQRT(EMOD/(SIGMANA*EULERN*FP));                                          LÆSC1120
148   1          EPSF=SIGMM/(EMOD*2E0);                                                           LÆSC1130
                      /* PROFILE POINTERS,INITIAL DESIGNAREAS & MOVELIMITS                     */LÆSC1140
149   1          GET LIST(LPROF,DA,FM);                                                           LÆSC1150
150   1          AMAX=400E-4;             AMIN=0E0;                                               LÆSC1160
                      /* MINIMUM AREA FOR MAXIMUM SLENDERNESS                                  */LÆSC1170
152   1          IF AUTOVNF=0 THEN LIMAX=250; ELSE LIMAX=500;                                     LÆSC1180
155   1          DO IELM=1 TO NELM;       I=LGRUP(IELM);                                          LÆSC1190
157   1     1    IF LPROF(I)=4 THEN                                                               LÆSC1200
158   1     1    AMIN(I)=MAX(AMIN(I),(SL(LENS(IELM))/LIMAX)**2/ALPHA);                 END;       LÆSC1210
160   1          DO I=1 TO NGRUP;         /* AMIN,AMAX OG AKTIVE VARIABLE                      */LÆSC1220
161   1     1    IF FM(I)=0E0 THEN        AMIN(I),AMAX(I)=DA(I);  ELSE                            LÆSC1230
163   1     1    DO ;                     L=LPROF(I);                                             LÆSC1240
165   1     2    IF L=1 THEN DO;          AMIN(I)=[illegible]E-4;   AMAX(I)=156.E-4;  END;         LÆSC1250
170   1     2    IF L=2 THEN DO;          AMIN(I)=2[illegible]E-4;  AMAX(I)=347.E-4;  END;         LÆSC1260
175   1     2    IF L=3 THEN              AMIN(I)=2[illegible]E-4;                                 LÆSC1270
177   1     2    NAGRUP=NAGRUP+1;         LAGRUP(NAGRUP)=I;                                       LÆSC1280
179   1     2    END; END;                                                                        LÆSC1290
                      /* GRAD. OF MERIT                                                        */LÆSC1300
181   1          GRAD=0E0;                                                                        LÆSC1310
182   1          DO IELM=1 TO NELM;       IG=LGRUP(IELM);          IE=LENS(IELM);                 LÆSC1320
185   1     1    IF FM(IG)¬=0E0 THEN      GRAD(IG)=GRAD(IG)+SL(IE);    END;                       LÆSC1330
                      /* INIT AF EIGENVECTORS                                                  */LÆSC1340
188   1          IF ANSVING=1 THEN DO;                                                            LÆSC1350
190   1     1    PIH=2E0*ATAN(ONE);       K=DOFNO*NHORZ+1;                                        LÆSC1360
192   1     1    DO I=1 TO NVERT;         KSI=FLOAT(I)/FLOAT(NVERT);                              LÆSC1370
194   1     2    DO J=0 TO NHORZ;         K=K+DOFNO;                                              LÆSC1380
196   1     3    DO IEV=1 TO NEV;         FAC=-1+2*IEV;                                           LÆSC1390
198   1     4    EV(IEV,K)=SIN(FAC*PIH*KSI);                                                      LÆSC1400
199   1     4    END; END;  END;  END;                                                            LÆSC1410
203   1          RETURN;                                                                          LÆSC1420
204   1          END LÆS;                                                                         LÆSC1430
```

```
STMT LEVEL NEST
  1              MASS:      /* LUMPED MASS/DISPLACEMENT & LUMPED MASS/NODE              */MASS0010
  2     1        PROCEDURE(MILIEUB1,ELEMTOPO,ELEMGEO,AREA,M,WM);                          MASS0020
                 DECLARE ((DOFNO,NDIS,NELM) EXTERNAL,                                      MASS0030
                         IELM,IGRUP,IENS,IA,IB,I,J) BIN FIXED,                             MASS0040
                         (B2,(AREA,M,WM)(*)) BIN FLOAT(53),                                MASS0050
                         1 MILIEUB1,                                                       MASS0060
                           2 (PJ,BNO,GAMMA)(*) BIN FLOAT(53),                              MASS0070
                         1 ELEMTOPO,                                                       MASS0080
                           2 ((LGRUP,LENS,LKE,LNSNIT,LPRUF)(*),TOPO(*,*)) BIN FIXED,       MASS0090
                         1 ELEMGEO,      2 (SL(*),A(*,*)) BIN(53);                         MASS0100
  3     1        M=BNO;                                                                    MASS0110
  4     1        DO IELM=1 TO NELM;      IENS=LENS(IELM);       IGRUP=LGRUP(IELM);         MASS0120
  7     1    1   IA=DOFNO*TOPO(IELM,1);                         IB=DOFNO*TOPO(IELM,2);     MASS0130
  9     1    1   B2=-(PJ(IELM)*A(IENS,1)+GAMMA(LPROF(IGRUP))*AREA(IGRUP)*SL(IENS))/2E0;   MASS0140
 10     1    1   M(IA-1)=M(IA-1)+B2;     M(IB-1)=M(IB-1)+B2;                               MASS0150
 12     1    1   END;                                                                      MASS0160
 13     1        I=0;                                                                      MASS0170
 14     1        DO J=2 BY 3 TO NDIS;    I=I+1;                                            MASS0180
 16     1    1   M(J-1),M(J),WM(I)=- M(J)/9.81E0;                      END;                MASS0190
 18     1        END MASS;                                                                 MASS0200
```

```
STMT LEVEL NEST
  1                MFSB:     /* CHOLESKY DECOMPOSITION OF A SYMM. POS.DEF. BANDMATRIX   */MFSB0020
                   PROCEDURE(A,N,NUD);                                                    MFSB0030
  2     1          DECLARE ERROR CHAR(1) EXTERNAL,                                        MFSB0040
                           (I,J,ID,JEND,K,KK,KEND,LN,LNUD,M,N,NC,NR,NUD) BIN FIXED,       MFSB0050
                           (EPS,SUM,PIV,A(*,*)) BIN FLOAT(53);                            MFSB0060
  3     1          LN=N;                  LNUD=NUD;                                       MFSB0070
  5     1          NR=LN-LNUD;            NC,JEND=LNUD+1;                                 MFSB0080
  7     1          EPS=1E-16;             ERROR='P';                                      MFSB0090
  9     1          DO I=1 TO LN;          IF I>NR THEN JEND=JEND-1;                       MFSB0100
 12     1     1    KEND=NC;               M=NC-I;                                         MFSB0110
 14     1     1    IF M>0 THEN KEND=KEND-M;                                               MFSB0120
 16     1     1    IF A(I,1)<1E20 THEN                                                    MFSB0130
 17     1     1    DO J=1 TO JEND;        ID=J-1;                  KK=I;                  MFSB0140
 20     1     2    SUM=0E0;                                                               MFSB0150
 21     1     2    DO K=J+1 TO KEND;      KK=KK-1;                                        MFSB0160
 23     1     3    IF A(KK,1)<1E20 THEN                                                   MFSB0170
 24     1     3    SUM=SUM+A(KK,K)*A(KK,K-ID);                                            MFSB0180
 25     1     3    END;                                                                   MFSB0190
 26     1     2    SUM=A(I,J)-SUM;                                                        MFSB0200
 27     1     2    IF J=1 THEN IF SUM>0E0 THEN                                            MFSB0210
 29     1     2    DO;                                                                    MFSB0220
 30     1     3    IF SUM<=ABS(EPS*A(I,J)) THEN ERROR='W';                                MFSB0230
 32     1     3    PIV,A(I,J)=SQRT(SUM);                                                  MFSB0240
 33     1     3    END;                   ELSE                                            MFSB0250
 34     1     2    DO;                    ERROR='S';               GOTO RETUR; END;       MFSB0260
 38     1     2    ELSE                                                                   MFSB0270
 38     1     2    A(I,J)=SUM/PIV;                                                        MFSB0280
 39     1     2    IF J<=M THEN KEND=KEND+1;                                              MFSB0290
 41     1     2    END; END;                                                              MFSB0300
 43     1          ERROR='0';                                                             MFSB0310
 44     1          RETUR:                                                                 MFSB0320
                   END MFSB;                                                              MFSB0330
```

```
STMT LEVEL NEST
  1               MTDSBX:    /* CHOLESKY SOLUTION WITH SEVERAL RHS. STORED ROWWISE        */MTDSBX10
  2     1         PROCEDURE(A,R,N,NUD,MN,M);                                                  MTDSBX20
                  DECLARE (ERROR EXTERNAL,COPT INITIAL('0')) CHAR(1),                         MTDSBX30
                          (I,ISTA,IEND,INCR,J,K,KEND,KI,KINC,                                 MTDSBX40
                          KK,L,LM,LN,LNUD,LMN,MN,M,N,NC,NR,NUD) BIN FIXED,                    MTDSBX50
                          (SUM,H,(A,R)(*,*)) BIN FLOAT(53);                                   MTDSBX60
  3     1         LN=N;                 LNUD=NUD;                LMN=MN;                      MTDSBX70
  6     1         LM=M;                 ERROR='P';                                            MTDSBX80
  8     1         NC=LNUD+1;            NR=LN-LNUD;                                           MTDSBX90
 10     1         ISTA,INCR=1;          IEND=LN;                 KINC=-1;                     MTDSB100
 13     1         MAIN:                                                                       MTDSB110
                  DO I=ISTA TO IEND BY INCR;                                                  MTDSB120
 14     1    1    H=A(I,1);                                                                   MTDSB130
 15     1    1    IF H=0 THEN                                                                 MTDSB140
 16     1    1    DO;                   ERROR='S';               GOTO RETUR; END;             MTDSB150
 20     1    1    KEND=NC;                                                                    MTDSB160
 21     1    1    IF INCR=1 THEN L=NC-I; ELSE L=I-NR;                                         MTDSB170
 24     1    1    IF L>0 THEN KEND=KEND-L;                                                    MTDSB180
 26     1    1    IF H<1E20 THEN                                                              MTDSB190
 27     1    1    DO J=LMN TO LM;       SUM=R(J,I);              KI,KK=I;                     MTDSB200
 30     1    2    DO K=2 TO KEND;       KI=KI+KINC;              KK=KK-INCR;                  MTDSB210
 33     1    3    SUM=SUM-A(KI,K)*R(J,KK);                                                    MTDSB220
 34     1    3    END;                                                                        MTDSB230
 35     1    2    R(J,I)=SUM/H;                                                               MTDSB240
 36     1    2    END;                                                                        MTDSB250
 37     1    1               ELSE                                                             MTDSB260
 37     1    1    DO J=LMN TO LM;       R(J,I)=0E0;              END;                         MTDSB270
 40     1    1    END;                                                                        MTDSB280
 41     1         IF COPT='1' THEN                                                            MTDSB290
 42     1         DO;                   ERROR='0';               GOTO RETUR; END;             MTDSB300
 46     1         UPPER:                                                                      MTDSB310
                  COPT='1';             ISTA=LN;                 INCR=-1;                     MTDSB320
 47     1         IEND=1;               KINC=0;                                               MTDSB330
 49     1         GOTO MAIN;                                                                  MTDSB340
 51     1         RETUR:                                                                      MTDSB350
 52     1         END MTDSBX;                                                                 MTDSB360
```

```
STMT LEVEL NEST
  1                  PLOTPR:     /* PLOT CF 2-DIM STRUCTURE ON CALCCMP 563                          */FLOTPR10
                     PROCEDURE(MAL,ELEMTOPO,ELEMGEO,X1,X2);                                          FLOTPR20
  2    1             DECLARE (NELM,NBEL,KNC,NSNIT,NKRAFT) BIN FIXED EXTERNAL,                        FLOTPR30
                             (IA,IB,ICA,ICE,IELM,ISNIT,IKE,IDEL,IENS,IGRUP,LNBEL,                    FLOTPR40
                             UDBØJ INIT(1),FORSTØR,MAL) BIN FIXED,                                   FLOTPR50
                             IMAX BIN FIXED(31),                                                     FLOTPR60
                             HOVED CHAR(100) EXTERNAL,                                               FLOTPR70
                             (KSI,KSI1,L,V1,V2,VN,VF,VA,VB,Z1,Z2,SINFI,COSFI,DN,KL,AKL,              FLOTPR80
                             A1,A2,C0,C1,C2,C3,LI,PD,                                                FLOTPR90
                             (X1,X2)(*)) BIN FLOAT(53),                                              FLOTP100
                             1 MILIEUB(*),  2 (PJ,BNO,GAMMA)(*) BIN FLOAT(53),                       FLOTP110
                             1 ELEMGEO,     2 (SL(*),A(*,*),VNF(*)) BIN FLOAT(53),                   FLOTP120
                             1 ELEMTOPO,    2 (LGRUP,LENS,LKE,LNSNIT,LPROF)(*) BIN FIXED,            FLOTP130
                                            2 TOPO(*,*) BIN FIXED,                                   FLOTP140
                             1 STIVHED,     2 ((AREA,IMOM)(*),EMOD) BIN FLOAT(53),                   FLOTP150
                             1 RESPONS(*),  2 (X,KLA,DGA)(*) BIN FLOAT(53),                          FLOTP160
                             (XSTART INIT(0),YSTART INIT(5),FACTR) DEC FLOAT(6),                     FLOTP170
                             (XL,YL)(NSNIT) DECIMAL FLOAT(6),                                        FLOTP180
                             AFFIN      ENTRY(DEC,DEC,DEC,DEC),                                      FLOTP190
                             KURVE      ENTRY((*) DEC,(*) DEC,DEC,DEC,DEC,DEC,BIN FIXED(31),         FLOTP200
                                              BIN FIXED(31),DEC),                                    FLOTP210
                             PARALF     ENTRY(DEC,DEC),                                              FLOTP220
                             PLINE      ENTRY(DEC,DEC,DEC,DEC),                                      FLOTP230
                             PSTRING    ENTRY(DEC,DEC,DEC,CHAR(*),DEC,BIN FIXED(31));                FLOTP240
  3    1             UDBØJ=0;                                                                        FLOTP250
  4    1             PLOTFFU:    /* PLOT CF STRUCTURE & DISPLACED STRUCTURE CN CALCOMF 563 */FLOTP260
                     ENTRY(MAL,ELEMTOPO,ELEMGEO,X1,X2,FORSTØR,MILIEUB,RESPONS,STIVHED);              FLOTP270
  5    1             IF UDBØJ=1 THEN LNBEL=NBEL; ELSE LNBEL=1;                                       FLOTP280
  8    1             IMAX=20+LNBEL*25;                                                               FLOTP290
  9    1             IF IMAX>60 THEN CALL FLIM(IMAX);                                                FLOTP300
 11    1             CALL PARALF(10.,0.);                                                            FLOTP310
 12    1             CALL PSTRING(0.,0.,.3,SUBSTR(HOVED,1,80),0.,0);                                 FLOTP320
 13    1             CALL PSTRING(0.,.6,.3,SUBSTR(HOVED,81,20),0.,0);                                FLOTP330
 14    1             CALL PSTRING(0.,1.2,.3,'MAAL  1:'||SUBSTR(CHAR(MAL),7,3),0.,0);                 FLOTP340
 15    1             IF UDBØJ=1 THEN                                                                 FLOTP350
 16    1             CALL PSTRING(0.,1.8,.3,'FLYTNING X '||SUBSTR(CHAR(FORSTØR),7,3),0.,0);          FLOTP360
 17    1             CALL PARALF(XSTART,YSTART);                                                     FLOTP370
 18    1             FACTR=1E0/MAL;                                                                  FLOTP380
 19    1             CALL AFFIN(FACTR,0.,0.,FACTR);                                                  FLOTP390
 20    1             DO IDEL=1 TO LNBEL;                                                             FLOTP400
 21    1    1        DO IELM=1 TO NELM;                                                              FLOTP410
 22    1    2        IA=TOPO(IELM,1);          IB=TOPO(IELM,2);                                      FLOTP420
 24    1    2        CALL PLINE(X1(IA),X2(IA),X1(IB),X2(IB));                                        FLOTP430
 25    1    2        IF UDBØJ=1 THEN DO;                                                             FLOTP440
 27    1    3        IENS=LENS(IELM);          IGRUP=LGRUP(IELM);          IKE=LKE(IELM);            FLOTP450
 30    1    3        L=SL(IENS);               COSFI=A(IENS,1);            SINFI=A(IENS,2);          FLOTP460
 33    1    3        IDA=3*IA;                 IDB=3*IB;                                             FLOTP470
 35    1    3        V1=  COSFI*X(IDEL,IDA-2)+SINFI*X(IDEL,IDA-1);                                   FLOTP480
 36    1    3        VN=  COSFI*X(IDEL,IDB-2)+SINFI*X(IDEL,IDB-1) - V1;                              FLOTP490
 37    1    3        V2= -SINFI*X(IDEL,IDA-2)+COSFI*X(IDEL,IDA-1);                                   FLOTP500
 38    1    3        VF=(-SINFI*X(IDEL,IDB-2)+COSFI*X(IDEL,IDB-1) - V2)/L;                           FLOTP510
 39    1    3        VA= VF-X(IDEL,IDA);       VB=-VF+X(IDEL,IDB);                                   FLOTP520
 41    1    3        DN=VN+DGA(IDEL,IELM)+VNF(IELM);                                                 FLOTP530
 42    1    3        PD=(PJ(IDEL,IELM)+GAMMA(IDEL,LPROF(IGRUP))*AREA(IGRUP)*L*COSFI)                 FLOTP540
                         *L**2/(EMOD*IMOM(IGRUP));                                                   FLOTP550
 43    1    3        LI=L*SQRT(AREA(IGRUP)/IMOM(IGRUP));                                             FLOTP560
```

```
        /* PLOT OF 2-DIM STRUCTURE ON CALCOMP 563                    */FLCTPR1C                  PAGE

STMT LEVEL NEST

 44     1    3     KL=KLA(IBEL, IELM);          AKL=ABS(KL);                                                 FLCTP570
 46     1    3     IF NKRAFT=1 & AKL>.15 THEN                                                                FLCTP580
 47     1    3     CALL KLVIRKN(KL,A1,A2,'FLCTPR',LI,VA,VB,PD,IKE,C0,C1,C2,C3,VF,L,L);                       FLCTP590
 48     1    3                                   ELSE                                                        FLCTP600
 48     1    3     IF IKE=4 THEN                                                                             FLCTP610
 49     1    3     DO;          /* IN COMPRESSION ECCENTRICITY & INIT.CURVATURE IS INCLUD.*/                 FLCTP620
 50     1    4     IF KL>0E0 THEN                                                                            FLCTP630
 51     1    4     IF PD>=0E0 THEN C0=1E0; ELSE C0=-1E0;                   ELSE C0=CEC;                      FLCTP640
 55     1    4     PD=PD:C0*16E-3*KL*AKL;                                                                    FLCTP650
 56     1    4     VA,VB=PD/24E0+C0*KL*AKL/(40E0*LI)+C0*8E-3;                                                FLCTP660
 57     1    4     END;                                                                                      FLCTP670
 58     1    3     DO ISNIT=1 TO NSNIT;                                                                      FLCTP680
 59     1    4     KSI=FLOAT(ISNIT-1)/FLOAT(NSNIT-1);                      KSI1=1E0-KSI;                     FLCTP690
 61     1    4     Z1=FORSTOR*(V1+VN*KSI) + L*KSI;                                                           FLCTP700
 62     1    4     IF NKRAFT=1 & AKL>.15 THEN                                                                FLCTP710
 63     1    4     IF KL>0E0 THEN                                                                            FLCTP720
 64     1    4     Z2=FORSTOR*(-V2+L*(C1*SIN (KL*KSI)/KL-C2*(1E0-COS (KL*KSI))                               FLCTP730
                         +C3*KSI+C0*KSI**2/2E0));                                                            FLCTP740
 65     1    4                 ELSE                                                                          FLCTP750
 65     1    4     Z2=FORSTØR*(-V2+L*(C1*SINH(KL*KSI)/KL-C2*(1E0-COSH(KL*KSI))                               FLCTP760
                         +C3*KSI+C0*KSI**2/2E0));                                                            FLCTP770
 66     1    4                 ELSE                                                                          FLCTP780
 66     1    4     Z2=FORSTOR*(-V2-L*KSI*(VF-KSI1*(KSI1*VA+KSI*VB+KSI*KSI1*PD/24E0)));                       FLCTP790
 67     1    4     XL(ISNIT)= COSFI*Z1 + SINFI*Z2;                                                           FLCTP800
 68     1    4     YL(ISNIT)= SINFI*Z1 - COSFI*Z2;                                                           FLCTP810
 69     1    4     END;                                                                                      FLCTP820
 70     1    3     CALL KURVE(XL,YL,1.,1.,-X1(IA),-X2(IA),-1, 3,.3/FACTR);                                   FLCTP830
 71     1    3     END;                                                                                      FLCTP840
 72     1    2     END;                                                                                      FLCTP850
 73     1    1     CALL PARALF(23/FACTR,C.);                                                                 FLCTP860
 74     1    1     END;                                                                                      FLCTP870
 75     1          END FLOTPR;                                                                               FLCTP880
```

```
STMT LEVEL NEST
  1                 PROFIL:     /* MOMENT OF INERTIA & SECTION MUDULUS OF DIFF.PROFILES       */PROFIL10
                    PROCEDURE(AREA,IMOM,WMOM,LPROF,JNY,JGL);                                    PROFIL20
  2      1          DECLARE ((NGRUP,REANAL) EXTERNAL,L,I,JNY,JGL,LPROF(*)) BIN FIXED,            PROFIL30
                            ((ALPHA,BETA) EXTERNAL,                                             PROFIL40
                            A,EI,EW,FI,FW,(AREA,IMOM,WMOM)(*))BIN FLOAT(53);                     PROFIL50
  3      1          DO I=1 TO NGRUP;            A=AREA(I);                     L=LPROF(I);      PROFIL60
  6      1     1    IF REANAL=1 THEN IF ¬(I=JNY|I=JGL) THEN GO TO SLUT;                          PROFIL70
  9      1     1    IF L=1 THEN DO;             EI=2.31026E0;                  FI=3.42084E-4;    /*IPE*/PROFIL80
 13      1     2                                EW=1.65410E0;                  FW=1.50986E-3;END;      PROFIL90
 16      1     1            ELSE                                                                 PROFIL100
 16      1     1    IF L=2 THEN                                                           /*HEA*/PROFIL110
 17      1     1    IF A<=1.24E-2 THEN DO;  EI=2.25915E0;                  FI=1.41040E-4;    PROFIL120
 21      1     2                            EW=1.64062E0;                  FW=1.03990E-3;END; PROFIL130
 24      1     1                    ELSE                                                     PROFIL140
 24      1     1    IF A<=1.98E-2 THEN DO;  EI=2.84863E0;                  FI=1.24245E-4;    PROFIL150
 28      1     2                            EW=1.86951E0;                  FW=0.98994E-3;END; PROFIL160
 31      1     1                 ELSE DO;   EI=3.29956E0;                  FI=0.91308E-4;    PROFIL170
 34      1     2                            EW=2.04627E0;                  FW=0.87735E-3;END; PROFIL180
 37      1     1            ELSE                                                             PROFIL190
 37      1     1    IF L=3 THEN                                                       /*HEB*/PROFIL200
 38      1     1    IF A<=1.61E-2 THEN DO;  EI=2.31478E0;                  FI=1.02348E-4;    PROFIL210
 42      1     2                            EW=1.67956E0;                  FW=8.67322E-4;END; PROFIL220
 45      1     1                 ELSE DO;   EI=3.32365E0;                  FI=6.33018E-5;    PROFIL230
 48      1     2                            EW=2.08261E0;                  FW=7.15847E-4;END; PROFIL240
 51      1     1            ELSE DO;                                       /* GEOM.SIMILAR    */PROFIL250
 52      1     2                            EI=2.E0;                       FI=ALPHA*1E-4;    PROFIL260
 54      1     2                            EW=1.5E0;                      FW=BETA*1E-3;     END;  PROFIL270
 57      1     1    WMOM(I)=FW*(A*1.E2)**EW ;                                                PROFIL280
 58      1     1    IMOM(I)=FI*(A*1.E2)**EI ;                                                PROFIL290
 59      1     1    SLUT:                                                                    PROFIL300
                    END;                                                                     PROFIL310
 60      1          END PROFIL;                                                              PROFIL320
```

```
/* ALL PRINTOUTS EXCEPT FORCES AND FI                                               */SKRIV010                    PAGE
 STMT LEVEL NEST
   1            SKRIV:     /* ALL PRINTOUTS EXCEPT FORCES AND FI                                                */SKRIV010
   2      1     PROCEDURE;                                                                                      SKRIV020
                DECLARE HOVED EXTERNAL CHAR(100),                                                               SKRIV030
                        ((ANSVING,NNC,NELM,DOFNO,NDIS,NBEL,NGRUP,NAREST,NITERA,NPROF)                           SKRIV040
                        EXTERNAL,I,J,N,N1,NON1,ITERA,IDEL) BIN FIXED,                                           SKRIV050
                        (T,W,C,S,VS,FIMAX,                                                                      SKRIV060
                        (X1,X2,DA,DU,EL,FN,FI)(*),SKEMA(*,*)) BIN FLOAT(53),                                    SKRIV070
                        1 ELEMTOPO,                                                                             SKRIV080
                          2 ((LGRUP,LENS,LKE,LNSNIT,LPROF)(*),TOPO(*,*)) BIN FIXED,                             SKRIV090
                        1 MILIEU(*),                                                                            SKRIV100
                          2 BELASTNING,3 (PJ,BNO,GAMMA)(*) BIN(53),                                             SKRIV110
                          2 FLYTNING,  3 (LNCDIS(*) FIXED,UST(*) FLOAT(53)) BIN,                                SKRIV120
                        1 ELEMGEO,     2 (SL(*),A(*,*)) BIN(53),                                                SKRIV130
                        1 STIVHED,     2 ((AREA,INOM)(*),EMOD) BIN(53),                                         SKRIV140
                        1 STYRKE,      2 (WMOM(*),(SIGMAN,SIGMAB,LIG)(*,*)) BIN(53),                            SKRIV150
                                       2 (FL,FC,FP,EULERN) BIN FLOAT(53);                                       SKRIV160
   3      1     SKRIV3:    /* NODE- & ELEM.PROPERTIES,LOADS & STRENGTHS                                       */SKRIV170
   4      1     ENTRY(ELEMTOPO,ELEMGEO,X1,X2,MILIEU,C,STYRKE,STIVHED);                                          SKRIV180
   5      1     SIGNAL ENDPAGE(SYSPRINT);                                                                       SKRIV190
                PUT EDIT('COORDINATES OF THE NODES','NODE       X1       X2')                                   SKRIV200
                        (SKIP(2),A,SKIP,A)                                                                      SKRIV210
                        ((I,X1(I),X2(I) DO I=1 TO NNO))                                                         SKRIV220
                        (SKIP,F(5),2 F(6,2),X(6))                                                               SKRIV230
                        ('ELEMENTPROPERTIES',                                                                   SKRIV240
                        'ELEMENT START  END  LGRUP  LKE LNSNIT LPROF   LENS COSFI',                             SKRIV250
                        '    SINFI LENGTH')(SKIP,A,SKIP,A,A)                                                    SKRIV260
                      ((I,TOPO(I,*),LGRUP(I),LKE(I),LNSNIT(I),LPROF(LGRUP(I)),LENS(I),                          SKRIV270
   6      1     PUT EDIT('SP.GRAVITY',GAMMA(NBEL,NPROF)',GAMMA,                                                 SKRIV280
                        A(LENS(I),*),SL(LENS(I))DO I=1 TO NELM))(SKIP,8 F(6),3 F(7,3));                         SKRIV270
                        'SIGMAN(NBEL,NPROF)',SIGMAN,                                                            SKRIV290
                        'LIG(NBEL,NPROF)',LIG,                                                                  SKRIV300
                        'SIGMAB(NBEL,NPROF)',SIGMAB)                                                            SKRIV310
                        (4 (SKIP(2),A,(NBEL )(SKIP,(NPROF)(E(12,4)))))                                          SKRIV320
                        ('FL       FC       FP       EULERN    EMOD',FL,FC,FP,EULERN,EMOD)                      SKRIV330
                        (SKIP,A,SKIP,4 F(6,3),E(10,3))                                                          SKRIV340
                        ('NODAL LOADS')(SKIP(2),A);                                                             SKRIV350
   7      1     PUT SKIP(2);                                                                                    SKRIV360
   8      1     DO I=1 TO NBEL;             DO J=1 TO NDIS;                                                     SKRIV370
  10      1   2 IF BNO(I,J)¬=0. THEN                                                                            SKRIV380
  11      1   2 PUT EDIT('        BNO(',I,',',J,')=',BNO(I,J))(A,F(1),A,F(3),A,F(6,2));                          SKRIV390
  12      1   2 END;  END;                                                                                      SKRIV400
  14      1     PUT SKIP EDIT('ELEMENTLOADS,TOTAL FORCE')(A);                                                   SKRIV410
  15      1     PUT SKIP(2);                                                                                    SKRIV420
  16      1     DO I=1 TO NBEL;             DO J=1 TO NELM;                                                     SKRIV430
  18      1   2 IF PJ(I,J)¬=0.   THEN                                                                           SKRIV440
  19      1   2 PUT EDIT('        PJ(',I,',',J,')=',PJ(I,J))(A,F(1),A,F(3),A,F(6,2));                            SKRIV450
  20      1   2 END;  END;                                                                                      SKRIV460
  22      1     PUT EDIT('ALLOW.DISPLACEMENTS,(LENGTH OR RADIANS)')(SKIP(2),A);                                 SKRIV470
  23      1     PUT SKIP(2);                                                                                    SKRIV480
  24      1     DO I=1 TO NBEL;             DO J=1 TO NDIS;                                                     SKRIV490
  26      1   2 IF UST(I,J)<1E20 THEN                                                                           SKRIV500
  27      1   2 PUT EDIT('        UST(',I,',',J,')=',UST(I,J))(A,F(1),A,F(3),A,F(9,3));                          SKRIV510
  28      1   2 END; END;                                                                                       SKRIV520
  30      1     IF ANSVING=1 THEN PUT SKIP(2) EDIT('SEAUC C=',C)(A,F(6,2));                                     SKRIV530
  32      1     SKRIV7:    /* AREA,MOMENT OF INERTIA & SECTION MODULUS                                        */SKRIV550
                ENTRY(STYRKE,STIVHED);                                                                          SKRIV560
```

```
        SKRIV:      /* ALL PRINTOUTS EXCEPT FORCES AND FI                          */SKRIVC1C          PAGE

STNT LEVEL NEST

 33     1               SIGNAL ENDPAGE(SYSPRINT);                                                       SKRIV570
 34     1               PUT EDIT('AREA          INCM          WMCM',                                    SKRIV580
                                 (AREA(I),IMOM(I),WMOM(I) DO I=1 TO NGRUP))                             SKRIV590
                                 (SKIP,A,(NGRUP) (SKIP,3 E(10,2)));                                     SKRIV600
 35     1               RETURN;                                                                         SKRIV610
 36     1               OPDAT:      /* UPDATE DESIGN REPORT                                            */SKRIV620
                        ENTRY(FIMAX,FI,SKEMA,ITERA,STIVHED,DA,BL,BU,S,T,W,C,FM);                        SKRIV630
 37     1               DO I=1 TO NGRUP;                                                                SKRIV640
 38     1    1          IF(DA(I)=BL(I))|(DA(I)=BU(I)) THEN SKEMA(I,ITERA)=-AREA(I);                     SKRIV650
 40     1    1                                         ELSE SKEMA(I,ITERA)= AREA(I);      END;          SKRIV660
 42     1               FIMAX=0.;  DO I=1 TO NAREST; FIMAX=MAX(FIMAX,FI(I));       END;                 SKRIV670
 46     1               SKEMA(NGRUP+1,ITERA)=S  ;                                                       SKRIV680
 47     1               SKEMA(NGRUP+2,ITERA)=FIMAX;                                                     SKRIV690
 48     1               IF ANSVING=1  THEN DO;                                                          SKRIV700
 50     1    1          SKEMA(NGRUP+3,ITERA)=T;                                                         SKRIV710
 51     1    1          SKEMA(NGRUP+4,ITERA)=C*W*9.81*.05/T**.33333;    END;                            SKRIV720
 53     1               PUT EDIT('MOVELIMITS',FM)(R(F1));                                               SKRIV730
 54     1               RETURN;                                                                         SKRIV740
 55     1               SKRIV6:     /* PRINT DESIGN REPORT                                             */SKRIV750
                        ENTRY(SKEMA,ITERA);                                                             SKRIV760
 56     1               SIGNAL ENDPAGE(SYSPRINT);                                                       SKRIV770
 57     1               ITERA=MIN(NITERA,ITERA);                                                        SKRIV780
 58     1               N=ITERA+1;                  N1=ITERA;                                           SKRIV790
 60     1               PUT EDIT('DESIGN REPORT',                                                       SKRIV800
                                 'DOWN:GROUPNO.,AREA IN CM**2,RIGHT: ITERATIONNO.',                     SKRIV810
                                 (I DO I=0 TO ITERA))(SKIP,A,SKIP,A,SKIP,(N) F(5));                     SKRIV820
 61     1               PUT SKIP(2) ;                                                                   SKRIV830
 62     1               DO I=1 TO NGRUP;                                                                SKRIV840
 63     1    1          PUT EDIT(I,(SKEMA(I,J) DO J=0 TO N1))(SKIP,F(3),(N)(F(9,2,4))); END;            SKRIV850
 65     1               PUT EDIT(' S ',(SKEMA(NGRUP+1,J) DO J=0 TO N1))(R(F2))                          SKRIV860
                                 ('MFI',(SKEMA(NGRUP+2,J) DO J=0 TO N1))(R(F2));                        SKRIV870
 66     1               IF ANSVING=1 THEN DO;                                                           SKRIV880
 68     1    1          PUT EDIT(' T ',(SKEMA(NGRUP+3,J) DO J=0 TO N1))(R(F2))                          SKRIV890
 69     1    1                   (' V ',(SKEMA(NGRUP+4,J) DO J=0 TO N1))(R(F2)); END;                   SKRIV900
 70     1               RETURN;                                                                         SKRIV910
 71     1               SKRIVX:     /* NODAL DISPLACEMNETS FROM LOADCASE IBEL                          */SKRIV920
                        ENTRY(X1,IBEL);                                                                 SKRIV930
 72     1               J=5-DOFNO;                                                                      SKRIV940
 73     1               PUT EDIT('NODAL DISPLACEMENTS,LOADCASE ',IBEL)(SKIP(2),A,F(2))                  SKRIV950
                                 ((NON1,(X1(DOFNO*(NON1-1)+I) DO I=1 TO DOFNO)                          SKRIV960
                                 DO NON1=1 TO NNO))(SKIP,(J)(F(7),(DOFNO)E(16,6)));                     SKRIV970
 74     1               RETURN;                                                                         SKRIV980
 75     1               F2:FORMAT(SKIP(2),A,(N)(F(9,4)));                                               SKRIV990
 76     1               F1:FORMAT(SKIP(2),A,SKIP(2),(NGRUP)(E(10,2)));                                  SKRI1000
 77     1               END SKRIV;                                                                      SKRI1010
```

```
STMT LEVEL NEST
  1             SVINGAN:    /* NEV FIRST EIGENVALUES & -VECTORS BY AN ITERATIVE METHOD*/SVINGA10
  2     1       PROCEDURE(KK,MASSE,EV,T,NDIS,NBAND,NEV);                               SVINGA20
                DECLARE (NDIS,NBAND,NEV,I,J,K,ET,J1) BIN FIXED ,                      SVINGA30
                        (ALFA,EPS,(BETA,OMSQNY,OMSQGL)(NEV),EM(NDIS),(T,MASSE)(*),    SVINGA40
                        (KK,EV)(*,*)) BIN FLOAT(53);                                  SVINGA50
  3     1       ET=1;                                                                  SVINGA60
  4     1       DO IEV=1 TO NEV;        OMSQNY(IEV)=(6.28/T(IEV))**2;        END;     SVINGA70
  7     1       EPS=1E-4;               IEV=1;                      J1=2;             SVINGA80
 10     1       OMSQGL=OMSQNY+1E0;                                                     SVINGA90
 11     1       DO WHILE(ABS(OMSQNY(NEV)-OMSQGL(NEV))/OMSQGL(NEV) >EPS);               SVING100
 12     1    1  DO J=IEV TO NEV;        EV(J,*)=MASSE*EV(J,*);   END;                 SVING110
 15     1    1  CALL MTDSBX(KK,EV,NDIS,NBAND,IEV,NEV);                                SVING120
 16     1    1  DO J=J1 TO NEV;         EM=EV(J,*)*MASSE;                             SVING130
 18     1    2  BETA(J-1)=SUM(EV(J-1,*)**2*MASSE);                                    SVING140
 19     1    2  DO K=1 TO J-1;          ALFA=SUM(EM*EV(K,*))/BETA(K);                 SVING150
 21     1    3  EV(J,*)=EV(J,*)-ALFA*EV(K,*);                     END;   END;         SVING160
 24     1    1  DO J=IEV TO NEV;                                                       SVING170
 25     1    2  OMSQGL(J)=OMSQNY(J);                                                   SVING180
 26     1    2  OMSQNY(J)=1E0/SUM(ABS(EV(J,*)));                                      SVING190
 27     1    2  EV(J,*)=EV(J,*)*OMSQNY(J);                        END;                 SVING200
 29     1    1  IF ABS(OMSQNY(IEV)-OMSQGL(IEV))/OMSQGL(IEV)<EPS THEN                   SVING210
 30     1    1  DO;                     IEV=IEV+1;                J1=IEV;     END;     SVING220
 34     1    1  END;                                                                   SVING230
 35     1       DO IEV=1 TO NEV;        T(IEV)=6.28318E0/SQRT(OMSQNY(IEV));END;       SVING240
 38     1       END SVINGAN;                                                           SVING250
```

```
OVERLAY ET
INSERT LÆS,****LÆSA
INSERT INELDI,INEIOA,INEUPA,INEUPB,INEVCS
OVERLAY ET
INSERT ANALYS,*ANALYSA
OVERLAY TO
INSERT KEMAT,KEIKK,BEL
INSERT **KEMATA,**KEIKKA,****BELA
INSERT KONVERG,KONVERGA
OVERLAY TO
INSERT MFSB,***MFSBA
INSERT MTDS3X,*MTDS3XA
INSERT SVINGAN,SVINGANA
INSERT EQUBE,**EQUBEA
INSERT MASS,***MASSA
OVERLAY ET
OVERLAY TRE
INSERT EVALFI,*EVALFIA
INSERT DUALP,**DUALPA
OVERLAY TRE
INSERT PROFIL,*PROFILA
INSERT SKRIV,**SKRIVA
OVERLAY ET
INSERT AL01PL,AL02PL,AL03PL,AL06PL,AL07PL
INSERT AL11PL,AL12PL,*AL12PLA,AL14PL,AL15PL
INSERT PLOTON,PLTERR,*PLTERRA,PLTMS1,*PLTMS1A
INSERT PLEXIT,PLOTROR,PLOTCOS,TIMEP
INSERT AFFIN,SYMBOL
INSERT KURVE,**KURVEA
INSERT PLINE,**PLINEA
INSERT PLOTPR,*PLOTPRA
```