OPTIMISATION

AND

PLASTIC ANALYSIS

ROBERT ERNEST MARKS

Submitted in Partial Fulfilment
of the Requirements for the
Degree of Master of Engineering Science

UNIVERSITY OF MELBOURNE

1969

Ackowledgement

Almost fifty year since this thesis was completed, two of the most supportive mentors of mine are still alive: Professors Len Stevens and Ray Toakley. I repeat my debt of gratitude to them for this work.

— Robert Marks, Sydney, February 2018.

Errata

On page 29b, in Figure VII, the symbol A* should be a*.

On page 29b, in Figure VII, the value −1.414 in vector p should be −0.141.

SYNOPSIS

This report considers the development of computer programs to carry out plastic analysis and design, using the techniques of mathematical optimisation.

A survey of the literature dealing with plastic analysis and design is made. The theoretical bases of four computer programs are reviewed. The four programs are suited to two- or three-dimensional pin-jointed trusses. They carry out, respectively,

i. a load factor analysis;
ii. a deflexion analysis at any stage of loading up to collapse;
iii. a design for minimum weight under one or several loading cases;
iv. an efficient weight design under one or several loading cases, considering the self-weight of the structural members.

Use of the programs is explained with examples, and some results obtained from their use are discussed.

LIST OF FIGURES

SYMBOLS

a	vector of member areas
a*	vector of area ratios
A	reference area, ($\sigma_Y A$ is reference tensile yield force)
A	diagonal area matrix
A_r	diagonal area matrix corresponding to redundant members
A*	diagonal area ratio matrix
b	lower limit on estimated possible values of y
B_o	force transformation matrix $q = B_o Q + B_1 x$
B_1	" " "
c	number of loading cases
C	complementary energy
C	connexion matrix
C_-	submatrix of C, corresponding to the unloaded, unsupported joints
C_+	square augmented connexion matrix, extra rows opposite redundants
d	number of degrees of freedom of collapse mechanism
d	vector of member deformations
D	vector of joint displacements
E	Young's modulus
E	force transformation matrix
F	structure flexibility matrix
F*	"dimensionless" structure flexibility matrix
g	load vector ($g = G_o P$)
G	force transformation matrix
G_o	dimensionless force transformation matrix $p = G_o P + G_1 r$
G_1	" " " "
I	unit matrix
j	number of joints of pin-jointed truss
k	transformed load matrix
L	unit load matrix: specifies load ratios and joints of application

m	number of members of pin-jointed truss
m	vector of coefficients of actual load factor
M_i	flexural section of i th beam
M	diagonal member length matrix
n	number of members at yield force
p	vector of ratios of member forces to member tensile yield forces
P	dimensionless load matrix
P_i'	the i th correction load matrix
q	member force vector
Q	actual load vector
r	number of internal redundants (lost during collapse)
r	vector of ratios of redundant member forces to member tensile yield forces
R_{max}	vector of greatest tensile member loads due to the external loading cases only
R_{min}	vector of greatest compressive member loads due to the external loading cases only
s	dimensionless positive redundant vector
T	transformed unit matrix
u	vector of relative redundant displacements
U	dimensionless complementary energy
U'	proportional to dimensionless complementary energy
U''	" " " " "
U	transformed connexion matrix
v*	vector of member volumes
V	volume of structure
V^*	"dimensionless" volume of structure
V	submatrix of U
w	weight density of flexural members
w	determinate member force system vector
W	weight of structure
W_c	weight of minimum weight design (m.w.d.)
x	vector of redundant member forces

y	vector of ratios of redundant forces to reference tensile yield force
z	positive redundant member force ratio vector
α	ratio of compressive to tensile yield stress; power in $w \propto M^{\alpha}$
β	actual load factor for unit load vector
γ	vector of ratios of actual member strains to tensile yield strains
ε_y	tensile yield strain
λ	"dimensionless" load factor for unit load vector
λ_c	collapse load factor
σ_Y	tensile yield stress
Φ	null matrix
0	null vector
I	unit vector
ℓ	vector of member length

CONTENTS

CHAPTER I. INTRODUCTION

The advent of the digital computer and the development of Operations Research have revolutionised many aspects of engineering. In structural engineering their development has coincided with that of methods of design and analysis based on the plastic theory of ductile structures. These methods combine simplicity of formulation with a rationality in description of the behaviour of such structures.

One of the first applications in structural engineering of the mathematical optimisation techniques of Operations Research came with the realisation that some formulations of analysis based on plastic theory led to optimisation problems and it was not long before plastic analysis had been automated. A few years later plastic design was similarly treated.

In the analysis of ductile structures, the ideal rigid-plastic models of behaviour can be treated either from a kinematic or mechanism approach or from a static or equilibrium approach to obtain the plastic collapse loads. The former corresponds to a linear programming maximisation, and the latter to a linear programming minimisation.

The plastic theory of design and analysis is formulated in terms of the strength of the material only - it can be assumed to exhibit rigid-plastic behaviour. But the ductile materials which can be analysed and designed using plastic theory are elastic-plastic in behaviour and undergo elastic deflexions before collapse. These deflexions are not given by plastic theory but may well be critical. (Elastic analysis gives the deflexion behaviour of the structure while ignoring the reserves of strength inherent in redundant, ductile structures.)

However, assuming perfectly elastic-perfectly plastic behaviour, the elastic deflexions at collapse may be estimated, as long as the correct member force distribution is known.

In cases of complete or overcomplete collapse the correct distribution is given by the equilibrium and yield criteria of plastic theory. But when the collapse is partial, as is common in complex structures, the rigid-plastic distribution attained is not unique: compatibility must be considered in order to obtain the correct elastic-plastic distribution. Compatibility must be considered also if the deflexions are required before incipient collapse: either in the fully elastic region or in the elastic-plastic region after yielding has first occurred.

Compatibility can be considered by solving additional virtual work or slope-deflexion equations, but the correct force distribution is also that distribution which leads to the minimum structural complementary energy. This can be treated as a quadratic programming problem with linear yield constraints.

Design usually aims to achieve a minimum total cost of materials, construction and maintenance. If the cost can be expressed in terms of the independent design variables and if the behaviour constraints are quantifiable then solution for the minimum cost design (m.c.d.) is possible. This formulation is not at all easy. But formulation of the minimum volume or weight design (m.w.d.) problem is simple: using strength constraints only this is the plastic design problem. Deflexion constraints (i.e. elastic-plastic design) lead to non-linearities. In general one can either minimise weight for constant strength-plastic (linear), or minimise weight for constant strain energy-elastic (nonlinear).

But what is the relevance of m.w.d. to m.c.d? In some structures the total cost is more dependent on the cost of the materials in the structure, while in other structures the cost of fabrication and maintenance are far more costly than the materials. In large office buildings the materials cost is but a small part of the total cost and a saving of a few percent in structural material costs is insignificant overall: the time and effort spent in attempting to achieve m.w.d. may even offset the savings in materials cost.

For any but the simplest structures, then, m.w.d. is hardly a practical economic objective. But it can help in understanding the most (structurally) efficient way of supporting loads. In this sense the designer can use it as a guide in attempting more rational design of structures. Some might say that the aesthetic pleasure obtained in achieving the simplicity of the m.w.d. is justification enough.

This thesis describes the theory and operation of four programs dealing with elastic-plastic and rigid-plastic two- or three-dimensional pin-jointed trusses. The programs are for determining the collapse load factor, for obtaining the elastic-plastic deflexions at any loading to collapse, and for obtaining the minimum weight design under one or several loading conditions and taking into account the self-weight of the bars.

CHAPTER II. LITERATURE SURVEY

2.1 Plastic Analysis

Following work done in England in the thirties and forties searching for simple yet rational procedures of design, interest was aroused in the so-called plastic behaviour of ductile materials, in particular, of mild steel. VAN DEN BROEK (1948) published "The Theory of Limit Design" in 1948 and in the following year BAKER (1949) published the results of his group in England.

These studies were concerned with the flexural behaviour of rigid-jointed frameworks, but several assumptions had to be made in order to be able to obtain results. The assumption of elastic-perfectly plastic moment-curvature behaviour, characterised by a sharply defined yield point and no strain hardening, was the most important. Also, it was assumed that plastic "hinges" formed at points where the bending moment reached yield - these hinges were capable of large angles of rotation at constant moment of resistance: the plastic moment. The effect on the plastic moment of axial loads and shear forces was neglected, and the elastic deformations of the structure were assumed to have no effect on the equilibrium equations.

It was found that to obtain the collapse load factor, a complete elastic-plastic analysis was not necessary. Upper and lower bounds for the factor were easily obtained and these could be narrowed without much difficulty (HORNE (1950), GREENBERG & PRAGER (1951)). Apart from the simplicity of the technique, it was a more realistic approach to structural behaviour: elastic design, with onset of yielding anywhere in the structure as the failure criterion, disregarded the reserves of strength inherent in redundant, ductile structures (NEAL & SYMONDS (1950)).

The bounds on the collapse load are obtained from the following three theorems:

Static Theorem: if, for a given set of external loads Q_s, an internal stress distribution can be found that satisfies equilibrium and doesn't violate the condition that yield nowhere be exceeded, then $Q_s \leq Q_c$, the collapse load. Such a state of stress is known as statically admissible. This theorem provides a necessary and sufficient condition for the structure to carry the loads (HORNE (1950), GREENBERG & PRAGER (1951)).

Kinematic Theorem: if, for a given set of external loads, Q_k, a collapse mechanism can be found consistent with equilibrium requirements, then $Q_k \geq Q_c$, the collapse load (GREENBERG & PRAGER (1951), NEAL (1956)).

Uniqueness Theorem: if, for a given set of external loads Q, a collapse mechanism can be found consistent with an internal stress distribution that satisfies equilibrium and nowhere exceeds yield, then $Q = Q_c$ and the mechanism is the actual mechanism of collapse (HORNE (1950)).

Similar theorems have been established for the general case of solid bodies of perfectly plastic material (DRUCKER et al (1951)).

Early workers were more interested in design than analysis, but failing a direct method of design, they developed iterative analysis. Methods were at first based on trial-and-error, but later one or both of the first two theorems:

HEYMAN (1951) (1) proposed a trial-and-error method but this was not suitable for partial collapse;

NEAL & SYMONDS (1952) suggested the "method of combining mechanisms" based on a kinematic approach, suitable for partial collapse;

GREENBERG & PRAGER (1951) suggested a method of upper and lower bound approaches successively, but this was awkward for partial collapse;

BAKER et al. (1956) suggested a trial-and-error approach;

NEAL & SYMONDS (1950) suggested a "method of inequalities" based on a static approach but this was tedious for complex structures;

Heyman and Nachbar suggested an alternating upper and lower bound approach but this required arbitrary cuts in the structure;

HORNE (1954) proposed a method of "plastic moment distribution", successively modifying the lower bound of the static approach;

HEYMAN (1968) described an extension of the method of combination of mechanisms to generate automatically a statically admissible bending moment distribution, but by hand only.

All these methods depended to some extent on the intuition and experience of the person making the calculations.

In 1951 CHARNES & GREENBERG (1951) showed that linear programming can be applied to the limit analysis of pin-jointed trusses. Their approach was based on the static theorem. They were able to develop systematic algebraic procedures for computation of collapse states, and to show that the kinematic approach leads to the dual problem.

FOULKES (1953), (1954), (1955) showed how the kinematic approach could be considered as a linear programming problem, and LIVESLEY (1956) analysed several frames by computer using a non-linear approach of the static theorem.

DORN & GREENBERG (1957) suggested that the equations

$$\beta \ L = C \ q \qquad ..(2.1)$$

$$-\alpha\sigma_Y \ A \ l \leq q \leq \sigma_Y \ A \ l \qquad ..(2.2)$$

(which ensure that the equilibrium and yielding conditions respectively are not violated) form the constraints of a limit analysis of a pin-jointed truss by the static approach: the greatest lower bound on β would be the actual collapse load factor β_c. Further, they suggested putting the bar forces in terms of the r redundant forces x:

$$\text{maximise} \quad [\,1 \;\vdots\; 0^T\,] \begin{bmatrix} \beta \\ x \end{bmatrix} \qquad ..(2.3)$$

$$\text{subject to} \begin{bmatrix} B_o L & \vdots & B_1 \\ -B_o L & \vdots & -B_1 \end{bmatrix} \begin{bmatrix} \beta \\ x \end{bmatrix} \leq \begin{bmatrix} A & 1 \\ \alpha A & 1 \end{bmatrix} \sigma_Y \qquad ..(2.4)$$

$$\text{where} \qquad q = \beta B_o L + B_1 x \qquad ..(2.5)$$

They also showed that the dual of this problem corresponds to the kinematic approach to the limit analysis problem. They mentioned an alternative problem which had non-negative variables and the equations 2.1 as explicit constraints.

CHARNES et al. (1959) extended the equivalence of dual linear programming problems with the static and kinetic plastic collapse principles to rigid-jointed frames, and showed, using virtical work, that a necessary and sufficient condition for collapse is that there is at least one solution to the static and at least one to the kinematic problems.

LIVESLEY (1964) showed how, for a rigid-jointed frame, the equilibrium equations could be modified so that the bending moments only were considered as significant. Elsewhere (LIVESLEY (1966), (1967)), he showed how the selection of redundants could be automated and how the collapse mode could be plotted by the computer. WRIGHT & BATY (1966) used Liversley's theory (LIVESLEY (1956)) to obtain both limit analysis and minimum weight design by computer.

KOOPMAN & LANCE (1965) extended the linear programming approach of the lower bound or static method to continuous structures.

The above approaches were mainly concerned with rigid-perfectly plastic material. PRAGER (1959) showed that the yield limit of a rigid-perfectly plastic continuum coincides with the load-carrying capacity of the corresponding elastic-perfectly plastic continuum.

HEYMAN (1959) (1), after HEYMAN & PRAGER (1958), making alternate use of equilibrium and yield, and equilibrium and mechanism criteria, described a program to obtain the collapse load factor automatically.

TOCHER & POPOV (1962) described a method, suitable for both proportional and variable repeated loading conditions, similar to linear programming, but not giving a lower bound β directly. WANG (1966) described an automated elastic-plastic analysis, following the loading history, using the displacement method. JENNINGS & MAJID (1965) described a similar program, taking axial load effects on the plastic moment into account. DAVIES (1967) described a method similar to Tocher and Popov's, but allowing for frame instability, strain hardening, and hinge reversal.

KORN & GALAMBOS (1968) compared analyses of first and second order accuracy, with and without axial deformations. They found that the analyses of some frames were not accurate using first order terms: these frames have many hinges, and almost level load-deflexion characteristics.

2.2 An Energy Principle for Elastic-Plastic Structures

In 1909 HAAR & VON KARMAN (1909) stated their well-known principle: "in the analysis of an elastic-perfectly plastic structure, of all the stress distributions which satisfy the equilibrium and yield conditions, that which actually occurs is that which minimises the elastic strain energy" (SAATY & BRAM (1964)).

SYMONDS & PRAGER (1950) (1) were able to prove the principle for the condition that no temporary unloading of the bars of the pin-jointed truss occurred, and they later (SYMONDS & PRAGER (1950) (2)) spoke of minimising the "fictitious residual energy" corresponding to the "fictitious state of residual stress" reached if complete unloading were a fully elastic process. PRAGER (1959) later showed that the principle was true even if temporary bar unloading occurred, as long as there was no decrease in the load factor.

In discussing Symonds & Prager, CHARLTON (1951) pointed out that the Haar-Karman principle was a particular case of Engesser's principle of minimum complementary energy for non-linear elastic systems: "since energy is a mathematical concept, application of (Engesser's) principle is valid in the non-conservative plastic range provided that a given static loading is applied only." Elsewhere, (CHARLTON (1950), (1952)), he showed that Engesser's principle depended on "the conservation of complementary energy", which excludes gross geometric distortions.

WESTERGAARD (1942) had shown how Engesser's principle could be applied to elastic, non-linear structures to account for settlement, temperature gradients, and displacement boundary conditions.

MATHESON (1959) showed how all the energy principles of Castigliano and Engesser, described by WILLIAMS (1938) and CHARLTON (1950), (1952) were related. ARGYRIS & KELSEY (1960) showed that the principle of stationary complementary potential energy was a generalisation of Castigliano's principle of minimum strain energy: "for given forces, the complementary energy of total deformation and the complementary work are minimum when equilibrium and compatibility are satisfied."

DORN (1960) showed that the dual of the principle of minimum elastic strain energy for an elastic-perfectly plastic material was "of all elongations and displacements which are compatible, the actual ones are those which minimise the potential energy" (SAATY & BRAM (1964)) - a generalisation of the principle of minimum potential energy.

In applied mechanics there are two theories of plasticity: the flow theory and the deformation theory. In the latter the relations between instantaneous states of stress and strain are so postulated that, when the strain is given, the stress is uniquely determined, or vice versa: as this determination may not be unique in both directions the deformation theory is unsuitable for describing completely the plastic behaviour of a metal and should be replaced by the flow theory (PRAGER (1948), WASHIZU (1968)).

GREENBERG (1949) showed that the Haar-Karman principle in the deformation theory of Hencky was analogous to the principle of minimum stress rate intensity in the flow theory of Prandtl and Reuss. Assuming the Haar-Karman principle, Hencky obtained his stress-strain relation (elastic-perfectly plastic) as the Euler-Lagrange equations of the integral being minimised.

WASHIZU (1968) showed that the Haar-Karman principle implied an absolute minimum for proportional loading.

2.3 Elastic Deflexions at Incipient Collapse

In an early paper GREENBERG & PRAGER (1951) noted that a general, simple method for estimating the deformation of an elastic-plastic structure was needed. Soon after, KNUDSEN et al. (1953) summarised and compared the methods available:

i. numerical integration of the actual moment curvature curve gave good agreement, but was tedious and empirical;

ii. mathematical integration of the idealised moment - curvature curve (HRENNIKOFF (1948)) was reasonable but complicated;

iii. the curvature-area method neglected spread of hinges and gave inaccurate results;

iv. simple plastic theory, neglecting strain hardening, but considering plastic spread, gave reasonable results;

v. the "plastic hinge method", based on elastic-perfectly plastic behavio ur, was very simple and gave reasonable results.

The "plastic hinge method" had been developed by SYMONDS & NEAL (1951), (1952) and HORNE (1950). Assuming elastic-perfectly plastic behaviour of rigid-jointed frames, the formation of plastic hinges, and neglecting the effects of shear and axial forces and stability, the method was well suited to structures which collapsed completely. These structures were determinate at incipient collapse and the mode of collapse with the equilibrium conditions led to the moment distribution, using plastic analysis and statics only.

To determine the last hinge to form, one could either assume in turn that each was the last, the correct assumption leading to the greatest deformations, or one could assume any to be the last and collapse the structure further until all but one of the calculated rotations were in the same sense as their bending moments, the one with no plastic rotation being the last hinge to form (SYMONDS & NEAL (1952), HORNE (1950)).

HORNE (1950) explained that at the required deflexion there would be elastic continuity at the last hinge to be formed, while all other hinges would show rotations in the directions corresponding to their full plastic moments. If the assumption of a particular hinge to be the last were incorrect, some of the calculated rotations would be in the wrong sense. Further collapse in the correct mechanism would lead to all rotations being either of the correct sign or zero. One (or several) would be zero: the last to form. As the collapse deflexions had been increased to achieve this, the last hinge would be characterised by the largest deformations.

SYMONDS & NEAL (1952) noted that for an r times redundant structure there were three types of collapse behaviour, characterised in part by n, the number of hinges

i. complete collapse - $n = r + 1$ leading to a determinate structure at incipient collapse, and a collapse mechanism with one degree of freedom;

ii. overcomplete collapse - $n > r + 1$ leading to a determinate structure at incipient collapse, and a collapse mechanism with more than one degree of freedom;

iii. partial or incomplete collapse - $n < r + 1$ leading to a redundant structure at incipient collapse, and a collapse mechanism with one degree of freedom.

Overcomplete collapse occurred when two or more hinges formed simultaneously at incipient collapse, leading to a mechanism with several degrees of freedom. Groups of hinges in turn must be assumed to form last, to obtain the correct group to form last.

Partial collapse, Symonds & Neal pointed out, meant that the elastic moment distribution was not completely determined by the values of the moments at the plastic hinges together with the conditions of static equilibrium. They suggested using the principle of least work to minimise the strain energy of the frame, leading to the correct moments. This was a tedious process by hand.

NEAL (1956) suggested using slope-deflexion equations with the condition of elastic continuity at unhinged joints, and LEE (1958), extended by ODEN (1967), proposed using the conjugate beam approach to shorten Neal's method slightly. HODGE (1959) also suggested minimising elastic energy: even though the frame were partially plastic at collapse the principle could be used, since the work done in plastic rotation was independent of the redundants. HEYMAN (1961), after STEVENS (1960), suggested using virtual work to obtain the redundant moment distribution, and discussing Heyman's paper, GREGORY (1962) suggested that the virtual work approach was mathematically equivalent to the method of static complementary energy.

In a paper specifically on the problem of partial collapse, PERRONE & SOTERIADES (1965) underlined that the positions of hinges in the elastic-plastic structure occurred where suggested by the rigid-plastic moment distribution only if they satisfied continuity. In discussion, GURFINKEL (1965) pointed out that for proportional loading the correct elastic-plastic moment distribution was the solution to a constrained minimisation of elastic strain energy. LIND (1965) suggested "rotation distribution", analogous to moment distribution for certain cases. THODANI (1966) suggested using "Mohr's equation", a form of virtual work.

In 1956 NEAL (1956) discussed certain assumptions necessary for the calculation of deflexions at incipient collapse: HORNE (1948) had concluded that the idealisations of no plastic spread of hinges and no strain-hardening were valid as the two effects cancelled each other in deflexion calculations. Neal pointed out that a further assumption was that, having formed, no plastic hinge unload. HODGE (1959) stated that if any hinge once formed had unloaded, then the predicted deformation would be an overestimate.

NEAL (1956) stated that no limit analysis could show whether unloading had occurred or not, and that the only safe procedure was to trace the successive formation and rotation of hinges in a step-by-step analysis. FINZI (1957) showed not only that unloading might occur, but that in general it would.

BERTERO (1965), in discussing Perrone & Sateriades, showed that the simplification of the virtual work approach of HEYMAN (1961) and MARTIN (1962) could not be used when hinges occurred which were not involved in the collapse mechanism. This would happen in partial collapse and in hinge unloading, he said, but, for partial collapse, correct use of the virtual work approach of HEYMAN (1961), HORNE (1962), or MARTIN (1962), or of the slope-deflexion equations (SYMONDS & NEAL (1952)) would lead to the discovery of these isolated hinges and the correct deflexions.

2.4 Plastic Design

Design methods based on iterative analyses are described above. They were indirect, and HEYMAN (1951) (2) and FOULKES (1953) were quick to realise that plastic theory could supply direct methods of design. These were concerned with minimum volume or weight design (m.w.d.), if not because this was practicable, then because it offered an ultimate criterion with which to assess practical designs.

In 1904 MICHEL (1904) obtained sufficient conditions for pin-jointed trusses to be of minimum weight, independent of the stress-strain relationship. Also studying the problem of a single, proportional loading system, FOULKES (1953), (1954), (1955) wrote a series of papers on the m.w.d. of rigid-jointed frames. Using a geometric analogue of design, he was able to prove three necessary and sufficient conditions, (analogous to the mechanism, equilibrium, and yielding criteria of limit analysis), which the m.w.d. must fulfil:

i. Mechanism condition: the design must be capable of failing in a mechanism (a "Foulkes mechanism") such that, for every design section M_i, $\sum_i$ hinge rotation associated with the section design $M_i \propto \sum_i$ length associated with M_i;

ii. Work equation: the load factor of the mechanism which satisfies i. must be unity;

iii. Yield condition: there must not exist any other mechanism for the design with a load factor of less than unity.

From these three conditions, FOULKES (1954) was able to prove two bounding theorems on the m.w.d., analogous to the static and kinematic approaches to plastic analysis respectively:

1. Upper-bound theorem: if a design collapses in a mechanism with a load factor of unity, satisfying iii., then its weight is greater than or equal to that of the m.w.d. This is the "safe" approach.

2. Lower-bound theorem: if a design satisfies i. and ii. then its weight is less than or equal to that of the m.w.d. This is the "unsafe" approach.

DRUCKER & SHIELD (1956) obtained sufficient conditions for continuous, three-dimensional structures.

SVED (1954) showed that, for a single loading system, the m.w.d. of a pin-jointed structure is statically determinate, for elastic or plastic behaviour. DORN et al. (1964), in a study concerned with the configuration as well as the sections of the m.w.d. structure, and PRAGER (1965), considering the analogy between network flows and plastic analysis, showed this also.

KICHER (1966) and SHEU & PRAGER (1968) showed analytically that the m.w.d. of a large class of structures subject to a single loading will be fully stressed and statically determinate. It follows that the m.w.d. of multi-loaded structures which are either statically indeterminate, or have buckling modes depending on loadings, generally will not be fully stressed. Structures which collapse partially under a single loading are not m.w.d.

HEYMAN (1959) (2) considered the absolute m.w.d. of structures with members of varying cross-section and showed that these could have 50% less material than structures with uniform sections as members.

For rigid-jointed frames, the weight density is $w \propto M^{\alpha}$, where M is the plastic strength, and α a constant. (In practice, $\alpha \simeq 0.6$). Most methods of design take $\alpha = 1$ which simplifies the procedure, with reasonable accuracy. PRAGER (1956), considering the convex problem of $0 < \alpha < 1$, obtained necessary and sufficient conditions for such a m.w.d. MEGAREFS & HODGE (1963) showed how to overcome the problems of nonlinearity and vanishing members using a density function analogous to strain energy. PRAGER & SHIELD (1967) developed a general theory of plastic design with a convex density.

Methods of solution of m.w.d. for single loadings were developed, usually based either on the upper ("safe") or on the lower ("unsafe") bounded approach. The "safe" method is better suited to automatic solution, while the "unsafe" to manual solution. HOSKIN (1960) noted that, as the lower bound approach dealt entirely with kinetic qualities, it did not directly give the individual bar sections, although it gave the m.w.d. weight and collapse mode.

The geometric analogue method of FOULKES (1953), a lower bound approach, had been based on an examination of all possible collapse modes. Later FOULKES (1955) showed how the problem was equivalent to one of linear programming.

The "method of inequalities" of HEYMAN (1951) (2) was a "safe" method but was tedious to solve by hand, and HEYMAN (1953) later suggested a design method which alternated between the two approaches, successively reducing the bounds on the m.w.d. HEYMAN & PRAGER (1958) automated this method, claiming it to be more efficient than methods based on one approach only.

LIVESLEY (1956) automated an upperbound approach, solving the minimisation by a method of modified steepest descent. This was the first automated procedure. WRIGHT & BATY (1966) used this procedure, expressing the moments in terms of the external loads and a set of redundants.

HOSKIN (1960), by analogy with CHARNES & GREENBERG (1951), showed the upper and lower bound design approaches to be equivalent to dual linear programming problems. CHAN (1964) was able to show that this duality led to Michell's necessary and sufficient conditions for the m.w.d. of pin-jointed trusses.

TOAKLEY (1967), (1968) described an automated "safe" approach which he solved using the dual simplex algorithm of linear programming. He showed that, ignoring instability and gross geometric distortion, the rigid-plastic assumption is reasonable for elastic-plastic behaviour.

The problem of multi-load design is more difficult than single load design, and can lead to shakedown (not treated here). HEYMAN & PRAGER (1958) noted that for several loadings the size of calculation is doubled, tripled and so on.

PEARSON (1958) suggested using a load space, analogous to Foulkes' design space, to estimate the worst effect on each possible mechanism. LIVESLEY (1959) noted that there was no meaning in speaking of the "worst" system of loading, as the m.w.d. balanced the effects of a number of extreme loading states, each exciting a different mechanism.

SAVE & PRAGER (1963), considering moving loads (an infinite number of loading cases), suggested a superposition principle for single spans, although this is not generally suitable. SHIELD (1963) gave sufficient conditions for m.w.d. under multi-loading conditions. Using this theory, MAYEDA & PRAGER (1967) extended the method for one loading case of HEYMAN (1959) (2) to multi-loading conditions. PRAGER (1967) showed that the usual proof of the existence of a Foulkes mechanism as a necessary condition for m.w.d. is not applicable to multi-loading.

DORN et al. (1964) showed how for two loadings the number of constraints in the linear programming problem would be doubled and the number of design variables increased. Similarly for more loading cases.

WRIGHT & BATY (1966) suggested obtaining a "design envelope" for all loading cases and hence getting the m.w.d. This is a highly inefficient method.

CHAN (1967), (1968), considering the duality of the two bounds approaches to m.w.d., extended the necessary and sufficient conditions of single loading design (MICHELL (1904), FOULKES (1954)) to the multi-loading case.

2.5 Simplifying Assumptions

In 1951 SYMONDS & NEAL (1951) noted that the development of plastic methods of analysis could take two directions: simple hypotheses leading to elegant mathematical theories and better understanding, or detailed behaviour of members, connexions, and frames, leading to more realistic descriptions of behaviour.

This work has been based on an ideal model of actual behaviour in pin-jointed trusses in the hope that this will lead to a better understanding of the theories attempting to describe realistic behaviour.

A main assumption has been that the load-deflexion behaviour, in both tension and compression, is either rigid-perfectly plastic (Fig. I) or elastic-perfectly plastic (Fig. II). If the load capacity is sustained over a sufficient deformation plateau, the simple mechanisms may be combined into a collapse mechanism. For mild steel in tension the description is a good approximation (Fig. III), but for compression members it may not be valid.

NEAL (1950) noted that for a slender, pin-jointed strut which is perfectly straight and loaded axially, the axial deformation below the Euler critical load is proportional to the load (i.e., linear elastic). When the Euler load is reached, buckling occurs, and the lateral deflexion increases at constant load. Hence, as in ideal plastic behaviour, the axial deformation increases at constant load. But the buckling is purely elastic (for large slenderness ratios) and the energy stored during buckling is recoverable as, unlike elastic-plastic unloading, the strut unloads elastically at constant load.

SYMONDS & PRAGER (1950) (1) noted that if the compression member had the flat yield stress-strain curve of ideal plasticity, there would be instability at yield. HRENNIKOFF (1965) has shown how strain hardening provides for instability.

Elastic buckling, however, occurs only at uneconomically large slenderness ratios. STEVENS (1968) considered elastic-plastic instability in compression members (Fig. IV). Behaviour depends on slenderness ratio and degree of end fixity; very short members exhibit good ideal plastic behaviour, but long

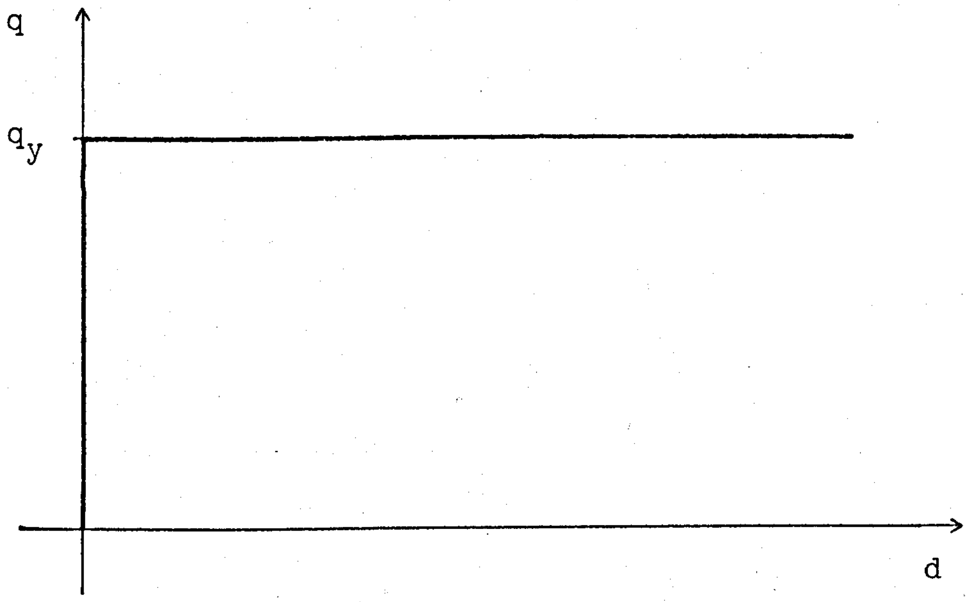


Figure I : Ideal rigid-plastic load-deformation behaviour.

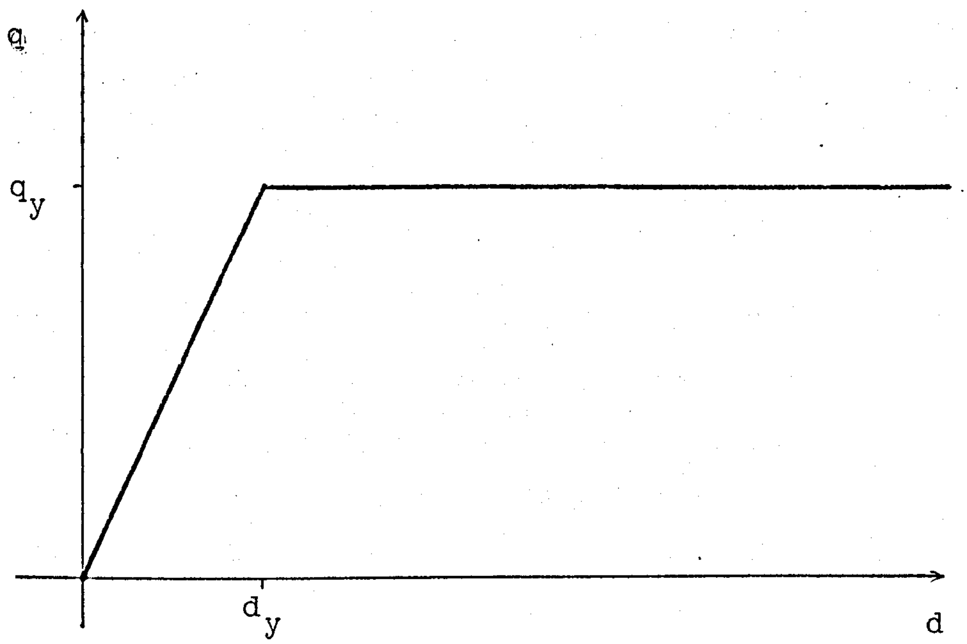


Figure II : Ideal elastic-plastic load-deformation behaviour.

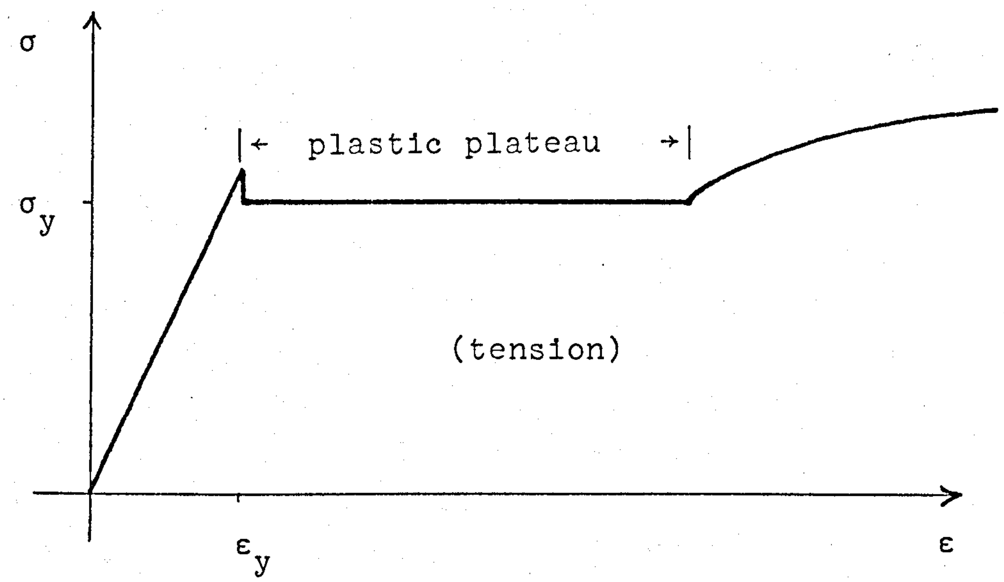


Figure III : Actual stress-strain relationship for mild steel.

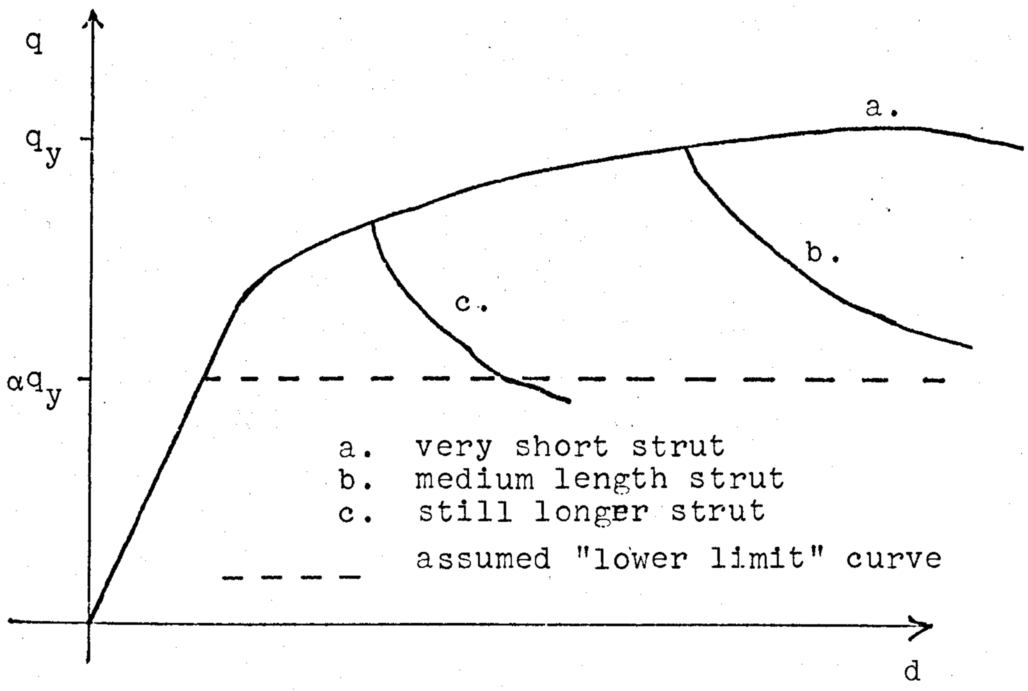


Figure IV : Actual load-deformation behaviour of pin-jointed struts, and the idealisation.

(from STEVENS (1968))

ones may have no plastic plateau at all.

This may be overcome by lowering the working yield load and using a load-deflexion diagram (Fig. IV) which generally gives a short plateau and leads to a conservative answer in analysis or design for reasonable axial strains. However, for small axial strains the approach may well be grossly overconservative, and, for the large axial strains sometimes found as the full strength of the structure develops, the approach is non-conservative.

Proportional loading is assumed: that is, the loading increases monotonically from zero. This does not ensure that no bar unloading occurs, but excludes failure by alternating plasticity or incremental collapse.

It is also assumed that the value of the tensile yield strain (that is, the ratio between Young's modulus and the tensile yield stress) is sufficiently small for the assumption that the deflexions before collapse have no effect on the equilibrium equations to be valid. That is, the analysis is concerned with small deflexions only.

In deflexion calculations, the further assumption that no bar unloading occurs is made. As noted in section 2.3, the only way to be sure that this doesn't happen is to analyse the complete loading history of the structure. PRAGER (1959) noted that the deflexions obtained using this assumption are still a good estimate, in fact are over-conservative.

CHAPTER III. GREATEST LOWER BOUND ESTIMATE OF COLLAPSE LOAD

3.1 Introduction

Several workers (CHARNES & GREENBERG (1951), FOULKES (1955), DORN & GREENBERG (1957)) have shown that the problem of obtaining the plastic collapse load for a structure can be viewed as a problem in linear programming. This chapter explains the theory of the static, or lower bound, approach to the problem for pin-jointed trusses.

To achieve generality it is desirable to work in dimensionless quantities. To this end, define

$$p_n = q_n/\sigma_Y A_n \qquad ..(3.1)$$

where p_n is the ratio of the actual force in the member n to the tensile yield force of that member

q_n is the actual force in member n

σ_Y is the tensile yield stress of the material

A_n is the cross-sectional area of member n.

Then, in matrix notation

$$q = \sigma_Y A p \qquad ..(3.2)$$

where $A_{ii} = A_i$, area of the i th member

$A_{ij} = 0,\ i \neq j$

Also define $\lambda = \beta/A\ \sigma_Y$ ..(3.3)

λ is the load factor for any given unit load, to give the actual load divided by the reference tensile yield force ($= A\ \sigma_Y$)

β is the load factor for any given unit load, to give the actual load

A is the reference cross-sectional area

If Q is the actual load matrix, and L the unit load matrix specifying the ratios, directions, and points of application of the loads, then

$$\beta L = Q \qquad ..(3.4)$$

Define a "dimensionless" load matrix, given by

$$P = \lambda L = Q/A\ \sigma_Y \qquad ..(3.5)$$

Then it is possible to work entirely in "dimensionless" quantities, the only data needed being C, L, and A*, where A* is the area ratio matrix $A^*_{ii} = A_i/A$

$A^*_{ij} = 0, \; i \neq j$

3.2 Explicit Equilibrium Constraint Equations

The static approach states that if equilibrium is everywhere maintained and the yield force is nowhere exceeded, the external loads in equilibrium with the internal loads are equal to or less than the collapse loads.

In matrix notation, express equilibrium equations as

$$\beta \, L = Q = C \, q \qquad ..(3.6)$$

where C is the connexion matrix, a force transformation matrix, generally rectangular, (square only if structure determinate), and comprising the direction cosines of the members.

Express the yield condition as

$$\alpha \sigma_Y \, A \, 1 \leq q \leq \sigma_Y \, A \, 1 \qquad ..(3.7)$$

where α is the ratio of compressive to tensile yield stress,

1 is the unit vector.

The greatest value of β satisfying 3.6 and 3.7 is the collapse load factor. To obtain a linear programming problem, the rows in 3.6 corresponding to load components are added and divided to get

$$\beta = m^T \; q \qquad ..(3.8)$$

where m is a coefficient vector. The remaining rows are expressed by

$$0 = C_{-} \; q \qquad ..(3.9)$$

where 0 is the null vector

C_{-} is a submatrix of C, corresponding to the unloaded, unsupported joint components.

The linear programming problem is to maximise β subject to 3.7 and 3.9. The independent variables are q, and if there is complete collapse the set of q will be unique; if there is partial collapse the set of q will not be unique, as compatibility will have to be considered to get the actual set of q.

In "dimensionless" quantities, substituting

3.2 and 3.3, the problem becomes

$$\text{maximise} \quad \lambda = m^T A^* p \qquad ..(3.10)$$

$$\text{subject to} \quad \alpha\, 1 \leq p \leq 1 \qquad ..(3.11)$$

$$\text{and} \quad 0 = C_- A^* p \qquad ..(3.12)$$

This problem has m independent variables, where there are m members, and the number of constraints is

$\leq 2m + 2(2j - 4)$ if two-dimensional

$\leq 2m + 2(3j - 7)$ if three-dimensional

where there are j joints. The large size of this problem leads to inefficient use of computer storage and time.

3.3 The Problem in Terms of the Redundant Forces

As an alternative to the approach of section 3.2, this section deals with a more efficient formulation of the problem. The forces in a redundant pin-jointed truss can be expressed in terms of the external loads and a set of redundants:

$$q = B_o Q + B_1 x = \beta B_o L + B_1 x \qquad ..(3.13)$$

where B_o and B_1 are force transformation matrices
x is a vector of redundant member forces.

Given the set of redundants, x, LIVESLEY (1964) has shown how to obtain 3.13 from the equilibrium equation 3.6: add extra rows to the connexion matrix, corresponding to a set of "releases" or redundant forces, to obtain

$$\begin{bmatrix} \beta L \\ \cdots \\ x \end{bmatrix} = C_+ q$$

where C_+ is a non-singular, square matrix

Then $$q = C_+^{-1} \begin{bmatrix} \beta L \\ \cdots \\ x \end{bmatrix}$$

multiplying, $$q = E \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix} \qquad ..(3.14)$$

where E, a transformation matrix, has k columns less than C_+^{-1},

where $k = 2j - 4$ (two dimensions)

$3j - 7$ (three dimensions)

Equation 3.14 is equation 3.13 after the multiplication of B_o L.

The linear programming problem is then

maximise $\beta = [1 \vdots 0^T] \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix}$ ..(3.15)

subject to $\alpha\sigma_Y A 1 \leq E \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix} \leq \sigma_Y A 1$ ..(3.16)

This problem has $r + 1$ independent variables, where there are r redundant members, $r = m-2j+3$ (two dimensions)

$= m-3j+6$ (three dimensions)

and the number of constraints is $2m$. This represents a great saving of storage and time over the previous method, especially for structures with low redundancy.

Analogous with equation 3.2, define vector r such that

$$x = \sigma_Y A_r r \qquad ..(3.17)$$

where r is the vector of the ratios of redundant member forces to their tensile yield forces

A_r is the area matrix corresponding to the set of redundants.

Then equation 3.13 becomes

$$p = G_o P + G_1 r = \lambda G_o L + G_1 r \qquad ..(3.18)$$

where $G_o = A^{*-1} B_o$ ..(3.18a)

and $G_1 = A^{*-1} B_1 A^*_r$

multiplying, $p = G \begin{bmatrix} \lambda \\ \cdots \\ r \end{bmatrix}$ ..(3.19)

The linear programming problem is

maximise $\lambda = [1 \vdots 0^T]\begin{bmatrix} \lambda \\ \cdots \\ r \end{bmatrix}$ ..(3.20)

subject to $\alpha 1 \leq G_1 \begin{bmatrix} \lambda \\ \cdots \\ r \end{bmatrix} \leq 1$ ..(3.21)

In the program described in section 3.5, r has not been used, but a quantity y, given by

$$y = x/A \sigma_Y \qquad ..(3.22)$$

that is, y is the actual redundant force in terms of the reference tensile yield force. The problem becomes

$$\left.\begin{array}{ll} \text{maximise} & \lambda = [1 \;\vdots\; 0^T] \begin{bmatrix} \lambda \\ \cdots \\ y \end{bmatrix} \\ \text{subject to} & \alpha \ddot{1} \leq [G_o \; L \;\vdots\; A^{*-1} \; B_1] \begin{bmatrix} \lambda \\ \cdots \\ y \end{bmatrix} \leq \ddot{1} \end{array}\right\} \quad ..(3.23)$$

3.4 Automatic Selection of Redundants

The method of the previous section (LIVESLEY (1964)) suffers from the fact that a set of redundants must be supplied to the program and must be consistent and reasonably well-conditioned, so that the cut structure is not a mechanism.

Algebraic procedures for automatic selection of redundants have been developed by DENKE (1965), ROBINSON (1966), and LIVESLEY (1966), (1967). In a survey of the literature, ROBINSON (1968) noted that the best set of redundants is that which leads to a "cut", determinate structure as close as possible to the indeterminate structure. To achieve this, knowledge of the applied loading system and the relative flexibilities of members is needed. DENKE (1965) showed how to consider the relative flexibilities by dividing columns of the connexion matrix C and multiplying rows of the force vector q, but this procedure is not used here as the author wished to develop a "modular" set of subroutines, suitable for both analysis and design.

The method described below is based, as are all of the algebraic methods, on the well-known Gauss-Jordan method of solving simultaneous equations. It is virtually identical to the methods of ROBINSON (1966) and LIVESLEY (1966). As well as selecting a consistent, well-conditioned set of redundants x, the method generates the force transformation matrices B_o and B_1 of equation 3.13, analogous to the matrices G_o and G_1 of equation 3.18.

The first stage of the process is the transformation of equation 3.6 from the form.

$$\beta \; I \; L = C \; q$$

where I is a unit matrix, to the equivalent form

$$\beta \; T \; L = U \; q$$

where U consists of a unit matrix and other columns: the structure is redundant, having more members than degrees of freedom at its joints, so the number of columns of C > number of rows of C. For each row in turn, the largest element in C is determined and the row in I and C normalised with respect to this element. Multiplies of this row of C and I are added to the other rows of C and I respectively, in such a way as to make all other elements in the column of the largest element equal to zero.

If C is of full rank, the process may be applied to all rows without a complete row of zeros in both C and I occurring. If such a row does occur, it corresponds to a dependent equilibrium equation and may be neglected. Thus I and C are transformed respectively into T and U, U having $(m - r)$ columns with a single *1* as the only non-zero element.

If the structure is a mechanism, one or more of the rows of U will be entirely zeros, while the corresponding row(s) of T will have one or more non-zero elements. The number of degrees of freedom of the mechanism equals the number of such rows.

The columns of U are rearranged to form

$$\beta \; T \; L = [I \;\vdots\; V] \begin{bmatrix} w \\ \cdots \\ x \end{bmatrix} \qquad ..(3.24)$$

where w is the vector of member forces associated with the columns of I : the determinate system

x is the vector of member forces associated with the columns of V: the redundant system.

Rearranging equation 3.24

$$w = \beta \; T \; L - V \; x$$

and $\quad x = \qquad I \; x$

or, combining,
$$\begin{bmatrix} w \\ \cdots \\ x \end{bmatrix} = \begin{bmatrix} T & \vdots & -V \\ \cdots & \cdots & \cdots \\ \Phi & \vdots & I \end{bmatrix} \begin{bmatrix} \beta \; L \\ \cdots \\ x \end{bmatrix} \qquad ..(3.25)$$

where Φ is a null matrix. Returning to the original order of q involves an interchange of the rows of the transformation matrix in equation 3.25, opposite to the interchange of columns of U made previously.

Thus is obtained the required result, equation 3.13

$$q = [B_o \vdots B_1] \begin{bmatrix} \beta \cdots L_x \end{bmatrix}$$

or knowing L, equation 3.14

$$q = E \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix}$$

If L is known from the outset, and if B_o or B_1 are not required, the process is simpler and less demanding of computer storage. Operations are made directly on L instead of I, and equation 3.24 becomes

$$\beta k = [I \vdots V] \begin{bmatrix} w \\ \cdots \\ x \end{bmatrix} \qquad ..(3.26)$$

where the vector k is equal to T L

Hence follows

$$\begin{bmatrix} w \\ \cdots \\ x \end{bmatrix} = \begin{bmatrix} k & \vdots & -V \\ \cdots & & \cdots \\ 0 & \vdots & I \end{bmatrix} \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix} \qquad ..(3.27)$$

and rearranging for correct order of member forces leads to equation 3.14.

$$q = E \begin{bmatrix} \beta \\ \cdots \\ x \end{bmatrix}$$

In the programs described below, the procedure is carried out on an augmented matrix. C is augmented to $[C \vdots I]$ or $[C \vdots -L]$. The matrices of equation 3.23 are obtained after the selection of the redundants and the generation of the matrices B_o and B_1.

The process of searching for the largest pivot in each row should ensure that the resulting set of redundants is well-conditioned.

When B_o and B_1 have been generated, substitution in equation 3.18a yields G_o and G_1, the "dimensionless" transformation matrices.

3.5 A Program for the Collapse Load Factor

This section briefly describes a program written to set up equations 3.23 for any two- or three-dimensional pin-jointed truss, and to solve the dual of this linear programming problem, using the two-phase Standard Simplex algorithm.

The program, known as RANK PLASTICITY OPTIMISATION, comprises eight subroutines:

NOINV - dimensions the arrays and calls the other subroutines in turn;

PLOPT 3 - reads the data cards describing the truss, and forms the connexion, load, and area matrices. The connexion and load matrices are augmented;

RANK - using the "Rank technique" described in section 3.4, this generates the transformation matrix of equation 3.14, isolates a consistent set of redundants, and determines the degree of redundancy or the number of degrees of freedom of the structure. The load matrix is known, so the procedure is that of equations 3.26 and 3.27;

FORMS - calculates $A^{*-1} B_o$ and $A^{*-1} B_1$;

DUALP 3 - sets up the equations 3.23, forms the dual of this linear programming problem, and calls KRANTE and KRSIMP. The subroutine was developed by Mr. D.W. Bennett of Melbourne University;

KRANTE and KRSIMP - these solve the dual problem using the two-phase Standard Simplex algorithm. They were developed by Dr. K. Reinschmidt of M.I.T.;

ANSWER - prints the collapse load factor and calculates and prints the collapse force distribution corresponding to the set of optimum redundants.

To fully describe the structure, the members and joints are numbered, the support joints numbered last. A set of coordinates is decided upon, and the joint positions assigned coordinates.

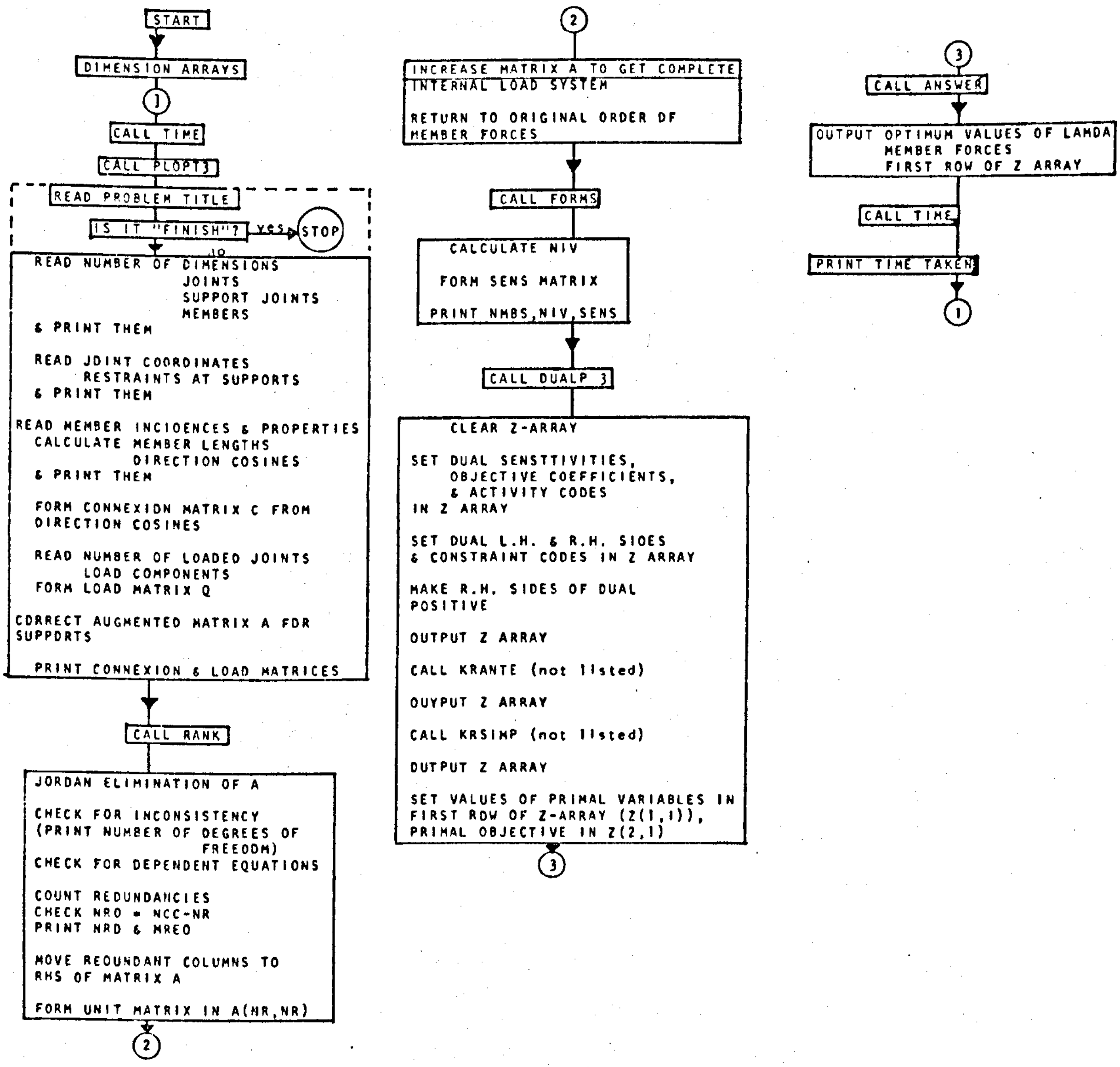


Figure V : Flow Diagram of RANK PLASTICITY OPTIMISATION

The data required for the analysis of a pin-jointed truss is, in order:

1. is the structure two- or three-dimensional? The number of joints, the number of supports, and the number of members;

2. the joint coordinates and the directions of restraint of the support joints;

3. the member cross-sectional areas and member incident joints;

4. the number of loaded joints;

5. the load components at each loaded joint. This method of specifying the structure (with a few modifications) is also used in the three programs described below.

A flow chart for the program described above is presented in Fig. V, and a full listing in Appendix A.

3.6 Analysis Examples

Three examples are presented: one to show the solution for a trivial case (truss 1), one to show the solution for a planar truss with varying cross-sectional areas (truss 2), and one to show the solution for a space truss (truss 3).

Truss 1 is a simple, once determinate planar truss with uniform cross-sectional areas. Clearly it must fail in either complete or over-complete collapse and so the force distribution obtained is unique, and identical in the rigid-plastic and elastic-plastic cases. The results can be seen in Fig. VI: the collapse load is $1.207 \times A \times \sigma_Y$.

Truss 2 is a four times redundant planar truss with the varying cross-sectional areas shown in Fig. VII with the joint and member numbering system. The collapse load is $1 \times A \times \sigma_Y$. The force distribution given by the analysis is not unique as the structure may collapse partially.

Truss 3 is a complex, three times redundant space truss with uniform cross-sections. The loading system, supports, and joint and membering systems are shown in Fig. VIII:

the structure has a triangular base and top, and each of its six joints is connected to the other five. The collapse load factor is 0.5000. As the structure may collapse partially, the rigid-plastic force distribution is not unique.

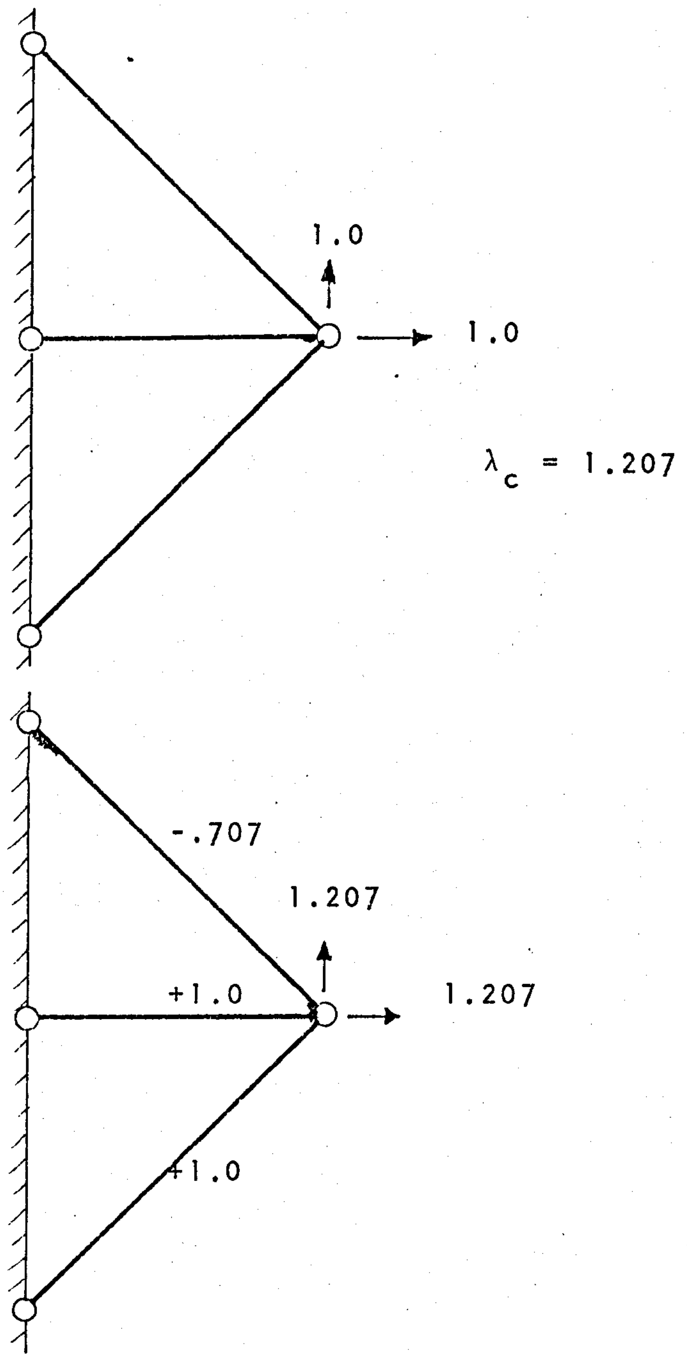
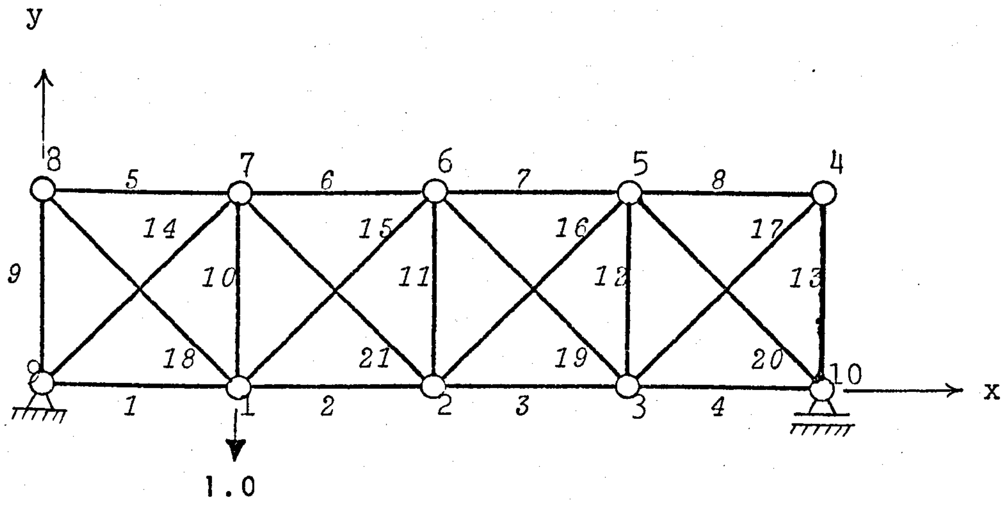


Figure VI : Collapse load factor analysis of truss 1.



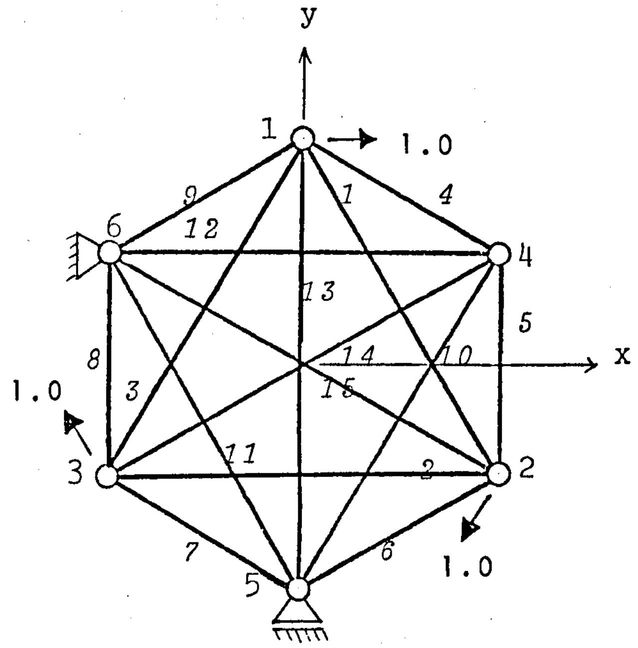
$$\lambda_c = 1.000$$

$$A^* = \begin{bmatrix} .750 & .500 & .750 & .500 & 5.00 & .750 & .500 \\ .250 & 5.00 & .750 & .250 & .750 & .250 & 1.061 \\ .354 & .354 & .354 & 5.00 & 5.00 & .707 & 5.00 \end{bmatrix}$$

$$p = \begin{bmatrix} 1.00 & 1.00 & 1.00 & 1.00 & 0.00 & -1.00 & 0.00 \\ 1.00 & 0.00 & 1.00 & 1.00 & 1.00 & 1.00 & -.999 \\ 1.00 & -1.00 & -.999 & 0.00 & -1.414 & -1.00 & 0.00 \end{bmatrix}$$

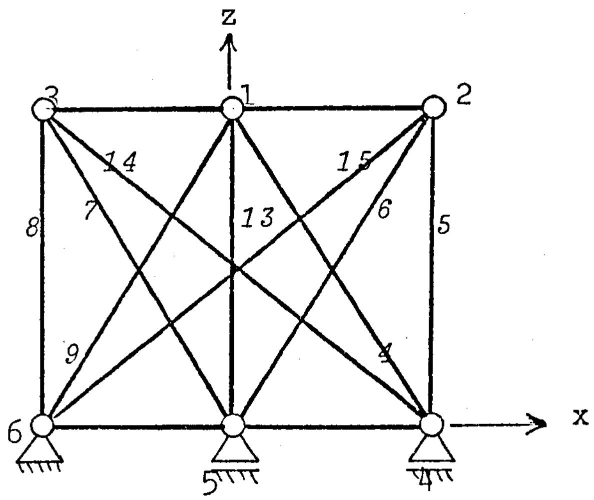
Figure VII : Collapse load factor analysis of truss 2.

PLAN:

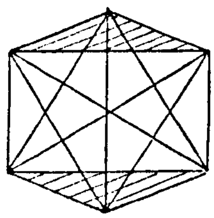


$\lambda_c = .500$

ELEVATION:



$$p = \begin{bmatrix} .500 & .207 & .207 & -.293 & .707 \\ 0.00 & .707 & -.293 & 1.00 & -.293 \\ -1.00 & .707 & -1.00 & -1.00 & -.586 \end{bmatrix}$$



SKETCH

Figure VIII : Collapse load factor analysis of truss 3.

CHAPTER IV. ELASTIC DEFLEXIONS AT INCIPIENT COLLAPSE

4.1 Introduction

Plastic design and analysis are essentially formulated in terms of the strength of structures, and may be considered as dealing with rigid-plastic models of behaviour. It has been shown (PRAGER (1959)) that the results obtained in dealing with the strength of rigid-plastic models are identical with those of elastic-perfectly plastic models.

But an elastic-perfectly plastic structure will experience elastic deflexions before collapse. The deflexions of actual ductile structures may well be critical in their performance under load. It would seem very useful to have a method of obtaining these deflexions.

In section 2.3 various methods were mentioned. The simplest of these was the "plastic hinge" method, and this gave reasonable results compared with more sophisticated methods. This section shows how this method, adapted for pin-jointed trusses, may be used to estimate the elastic-plastic deformations of such trusses at collapse and at any stage in the loading history before collapse.

The actual stress distribution can be found by minimising the complementary energy of the structure - for elastic-perfectly plastic behaviour, the elastic strain energy.

The simplifying (and restricting) assumptions have been discussed in section 2.5. The assumptions are proportional loading, neglect of changes of geometry on the equilibrium equations, a perfectly elastic-perfectly plastic load-deformation relationship for both tension and compression, and no unloading of yielded bars as the load is increased from zero to the collapse load.

The dimensionless notation introduced in section 3.1 will be used, although the deflexions will be in the same units as the joint coordinates.

4.2 Problems of Deflexion Calculations

A main problem is the assumption, mentioned above, that no unloading of yielded bars occurs as the load is increased from zero to the collapse load. This was discussed in sections 2.3 and 2.5: short of a complete loading history of the structure, it is reassuring that the assumption will lead to an over-estimate of deflexions.

In section 2.3 it was noted that an r times redundant structure could collapse in three basic ways:

i. complete collapse - the number of yielded members $n = r + 1$, leading to a determinate structure at incipient collapse and a collapse mechanism with one degree of freedom;

ii. over-complete collapse - the number of yielded members $n > r + 1$, leading to a determinate structure at incipient collapse and a collapse mechanism with more than one degree of freedom;

iii. partial collapse - the number of yielded members $n < r + 1$, leading to an indeterminate structure at incipient collapse and a collapse mechanism with one degree of freedom.

In the first and second cases the determinate set of forces at incipient collapse is unique and given by the procedure described in chapter III above. In the third case, considering rigid-plastic behaviour, the set of forces at incipient collapse is indeterminate and not unique. The set given by a load factor analysis, satisfying yield and equilibrium, will not necessarily satisfy the compatibility requirements of an elastic-plastic structure.

It must be added that, in complex structures, partial collapse is the most common type of collapse behaviour, although a combination of partial and over-complete collapse may occur if several bars yield together, forming a mechanism of more than one degree of freedom in one region of the structure, leaving the rest of the structure indeterminate.

In this case, n, the number of yielded bars, is no guide to the collapse behaviour, although the further knowledge of the number of degrees of freedom of the collapse mechanism or the degree of indeterminacy of the remaining frame will help.

If two independent mechanisms form at collapse of a structure, then even the additional information of number of degrees of freedom and the degree of indeterminacy may not indicate the actual behaviour. However, the assumption that a particular form of collapse occurs will be verified by the member deformations subsequently calculated.

4.3 Minimum Complementary Energy

The Haar-von Karman principle (as stated in section 2.2) of minimising the elastic strain energy is identical to Engesser's principle of minimum complementary energy, for a proportionally loaded, elastic-perfectly plastic structure.

The complementary energy C is given by

$$C = {}^{1}/_{2}\ q^{T}\ F\ q \qquad ..(4.1)$$

where F is the structural flexibility matrix $F_{ii}=L_i/E\ A_i$

$F_{ij}=0,\ i \neq j$

To use the dimensionless vector p, define

$$U = {}^{1}/_{2}\ p^{T}\ F^{*}\ p \qquad ..(4.2)$$

where F*is the "dimensionless" flexibility matrix

$$F^{*}_{ii} = A_i L_i / A$$

$$F^{*}_{ij} = 0,\ i \neq j$$

then $$U = C \times \frac{EA}{(\sigma_Y)^2}$$

Applying the principle is to minimise U

subject to $- \alpha\ 1 \leq p \leq 1 \qquad ..(4.3)$

and $\lambda\ L = P = C\ A^{*}\ p \qquad ..(4.4)$

the yield and equilibrium conditions respectively. This is with a definite load which is constant:

$\lambda\ L = P$

Using equation 3.18 to get p in terms of P and r

$$p = G_o P + G_1 r$$

where $G_o = A^{*-1} B_o$

$G_1 = A^{*-1} B_1 A^*_r$

thus $p = g + G_1 r$ ..(4.5)

and the problem becomes

minimise $U' = 2\, g^T F^* G_1 r + r^T G_1^T F^* G_1 r$

subject to $-\alpha l - g \leq G_1 r \leq + l - g$ ..(4.6)

Using a non-negative variable s, where

$$s = r + l$$ ..(4.7)

the problem becomes

minimise $U'' = 2(g^T - l^T G_1^T) F^* G_1 s + s^T G_1^T F^* G_1 s$

subject to $G_1 l - \alpha\, l - g \leq G_1 s \leq G_1 l + l - g$ ..(4.8)

This is a quadratic optimisation problem with linear constraints. As F^*, and hence $G_1^T F^* G_1$, is a symmetric, positive definite matrix, the function U'' is strictly convex. This means that the function has only one minimum. This is a global minimum. A program which minimises U'' subject to the constraints of equation 4.8 is described below.

4.4 Deflexion Analysis

The problems of calculating the elastic deflexions at incipient collapse and of determining which is the last bar to yield are closely linked. There are two ways of determining which is the last bar to yield: one may either assume in turn that each bar (or group of bars in over-complete collapse) is the last, the correct assumption leading to the greatest deflexions, or one may assume any bar to be the last, and collapse the structure further until all but one of the calculated plastic strains are in the same sense as their axial stresses, the bar with no plastic strain being the last to yield. The first method has been considered more suitable for automation and has been used in the program described in section 4.5 below.

Given the correct force distribution by minimising the complementary energy of the structure, the program counts the n yielded members, calculates the ${}^{n}Cr$ combinations of r different members (where r is the degree of redundancy lost in collapse), and forms elastic structures by eliminating in turn the columns or elements in the matrices C, ℓ, and p corresponding to the groups of plastic members assumed to yield first.

This, in effect, is reducing the elastic-plastic redundant structure to an elastic structure with constant loads replacing the first yielding members, determinate if failure by complete or overcomplete collapse, probably still redundant if failure by partial collapse. It is assumed that the $(n - r)$ remaining members at yield are the last to yield, that is, are not strained plastically at incipient collapse.

A set of elastic joint displacements and a set of elastic-plastic deformations of all members are calculated for each assumed elastic structure, as described below. The set of joint displacements with the largest overall values is the correct set and gives the correct group of members to yield last. The corresponding set of member deformations must have no elements of opposite sign to the corresponding elements in the set of member forces, p, and the assumed yielded members must have strains greater than the yield strain.

If the assumed elastic structure for any group forms a mechanism, then the group of $(n - r)$ members assumed last to yield is incorrect: the redundant structure must be stable until all n members have yielded.

The main drawback to automating this process completely is in determining the correct value of r: is the collapse partial or not? Consider the most general case of a highly redundant structure which collapses partially, several bars yielding together at final collapse.

If d, the number of degrees of freedom of the partial collapse mechanism, can be found, then r, the reduction in redundancy due to the partial collapse mechanism, is given by

$$r = n - d \qquad ..(4.9)$$

where n is the number of yielded members at collapse. This is so because as each member yields it reduces the overall redundancy by one, until the local degree of redundancy is zero, at incipient collapse. The number of degrees of freedom of the partial collapse mechanism is equal to the further number of bars which yield, forming the mechanism.

In general, then, the assumed elastic structure is still redundant, and the transformation matrices B_{oe} and B_{1e} can be generated from the connexion matrix C_e as described in section 3.3. (The subscript e implies that the structure is treated as wholly elastic; C_e is reduced from matrix C as mentioned above).

From the force method of analysis (LIVESLEY (1964)) the equations

$$D = B_{oe}^T \, F_e \, q_e \qquad ..(4.10)$$

$$u = B_{1e}^T \, F_e \, q_e \qquad ..(4.11)$$

are obtained, where D is the vector of joint displacements and u is the vector of relative redundant displacements and will equal zero if the structural member forces are compatible. (The subscript e implies that the columns or rows corresponding to the assumed group of r plastic members have been eliminated).

Non-dimensionally, that is, substituting

$$q_e = \sigma_Y \, A_e \, p_e \qquad ..(4.12)$$

into 4.10 and 4.11, the relationship became

$$\left(\frac{1}{\varepsilon_Y}\right) D = \left(\frac{E}{\sigma_Y}\right) D = B_{oe}^T \, \ell_e \, p_e \qquad ..(4.13)$$

$$\left(\frac{1}{\varepsilon_Y}\right) u = \left(\frac{E}{\sigma_Y}\right) u = B_{1e}^T \, \ell_e \, p_e \qquad ..(4.14)$$

where ℓ_e is the reduced vector of member lengths.

Then D and u can be evaluated for each assumed group and checked to find the overall largest corresponding to a zero u. A further check must be made of the member deformations by using

$$\left(\frac{E}{\sigma_Y}\right) d = C^T D \left(\frac{E}{\sigma_Y}\right) \qquad ..(4.15)$$

to evaluate the member deformations d of all members even those assumed to have yielded. Hence, γ_i can be calculated for each member, where γ_i is the ratio of actual strain to yield strain, given by

$$\gamma_i = \frac{\varepsilon_i}{\varepsilon_Y} = \frac{d_i}{\varepsilon_Y \ \ell_i} = \left(\frac{E}{\sigma_Y}\right) \frac{d}{\ell_i} \qquad ..(4.16)$$

If the member has yielded,

$$|\gamma_i| > 1 \qquad ..(4.17)$$

and $\quad \text{sign}\ (\gamma_i) = \text{sign}\ (p_i) \qquad ..(4.18)$

and if the member is one of a group to yield last,

$$|\gamma_i| = 1 \qquad ..(4.19)$$

Thus γ_i becomes an added check on the correct group of members last to yield.

In addition, γ_i is a measure of the size of plastic plateau required in order to develop the full strength of a ductile structure:

$$\text{plastic plateau} = \gamma_i - 1$$

This section has so far been mainly concerned with calculating the elastic deflexions at incipient collapse: this is the point of greatest load and largest deflexions before collapse, and for complete or over-complete collapse the actual force distribution can be calculated without recourse to compatibility considerations. But, using the Haar-von Karman principle, the actual force distribution can be determined for any pin-jointed truss at any stage of loading.

Hence, using the procedure outlined above of eliminating the elements of the structural matrices corresponding to members at yield, the elastic deflexions of the structure can be calculated at any stage of loading to collapse.

4.5 A Program to Calculate the Elastic Deflexions of Pin-Jointed Trusses Loaded to Collapse

This section describes briefly a program to calculate the compatible force distribution at any load to collapse, and to calculate the elastic deflexions of the structure and the elastic-plastic deformations of the members.

The programming problem of equation 4.8 is one of minimising a strictly convex quadratic function subject to linear constraints. KÜNZI et al. (1968) suggest using either of the direct methods of BEALE (1959) or of WOLFE (1959). However, the problem can be solved as an iterative linear programming problem, using the method of REINSCHMIDT et al. (1966).

Rather than work in the actual force hyper-space $V_r(s)$ they suggest working in the delta-force hyper-space $V_r(\Delta s)$, using piece-wise linearisation of the quadratic merit function U''. The problem was initially (equations 4.8)

minimise $\quad U'' = 2(g^T - l^T G_1^T)\ F{*}G_1 s + s^T G_1^T F{*} G_1 s$

subject to $\quad G_1 l - \alpha l - g \leq G_1 s \leq G_1 l + l - g$

or, more simply $\quad b \leq G_1 s \leq c$

Starting from an initial point s_o, the problem in delta-force hyper-space is

$$\text{minimise} \quad \Delta U'' = \left(\frac{\partial U''}{\partial s}\right)^T_{s_o} \Delta s$$
$$\text{subject to} \quad b - G_1 s_o \leq G_1 \Delta s \leq c - G_1 s_o \qquad ..(4.20)$$

that is,

$$\text{maximise} \quad \Delta U'' = -2(g^T + (s_o^T - l^T)G_1)F{*}G_1 \Delta s$$
$$\text{subject to} \quad G_1(l - s_o) - \alpha l - g \leq G_1 \Delta s \leq G_1(l - s_o) + l - g \qquad ..(4.21)$$

This gives optimum Δs, and the next point in the force hyper-space is given by

$$s_1 = s_o + \Delta s$$

and so on. Adaptive move limits are used to achieve unconstrained or semi-constrained optima. (Note that the substitution

4.7 for r was unnecessary, as the vector Δs may still have both positive and negative elements).

The program, known as MINIMUM COMPLEMENTARY ENERGY, comprises ten subroutines:

ENERGY - dimensions the variable arrays and calls the other subroutines in turn;

PLOPT 7 - reads the data describing the truss and forms the connexion, load, member length and area matrices;

RANK 2 - using the "Rank technique", this generates the transformation matrices B_o and B_1, isolates a consistent set of redundants, and determines the degree of redundancy and/or the number of degrees of freedom of the structure;

HAAR - forms the matrices G_1, P, and g, and prints their values. Forms s_o, the initial set of redundants;

KARMAN - forms the objective coefficients $\left(\frac{\partial U''}{\partial s}\right)_{s_i}$ and computes the left and right hand sides of the constraint equations 4.21, for any force point s_i;

DUALP 5 - forms the dual of the linear programming problem of equation 4.21 and calls KRANTE and KRSIMP;

NOWEND - prints the values of the number of iterations, $\Delta U''$, p, Δs_i, s, and the "dimensionless" complementary energy U;

DEFLN - reads the values of r and n and calculates the ${}^{n}C_{r}$ combinations. (It could have been written to count n, the number of yielded members, to form a reduced connexion matrix by eliminating all columns corresponding to yielded members, to calculate d, the number of degrees of freedom of the collapse mechanism, using RANK 2, and hence to obtain r, the loss of redundancy of the structure at collapse from equation 4.9, but there was insufficient time to automate the process fully. As described here, it is semi-automated only). Forms the assumed elastic structural matrices B_{oe}, B_{1e}, F_e^*, and q_e, checking for the stability of the assumed structure. Calculates D_e and d and checks that u is zero and computes γ_i, and prints these values.

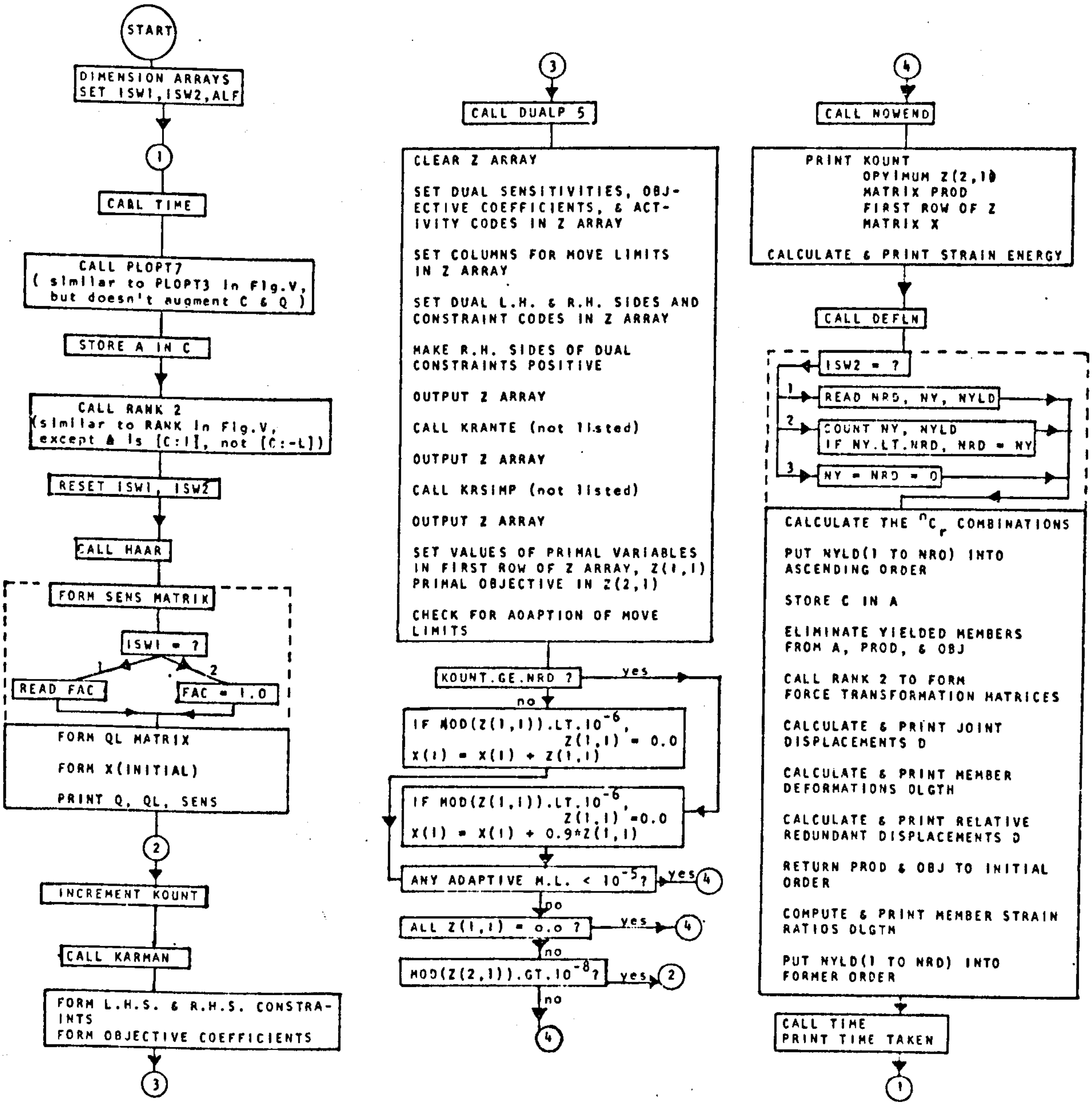


Figure IX : Flow Diagram of MINIMUM COMPLEMENTARY ENERGY

A flow diagram of this program is presented in Fig. IX, and a full listing can be seen in Appendix B.

4.6 Deflexion Calculation Examples

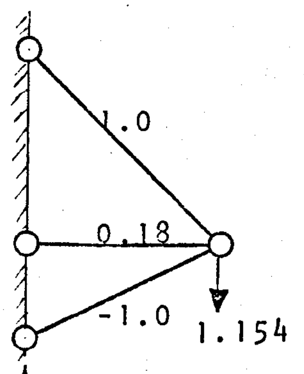
Three examples are presented: a trivial case (truss 4), a planar truss with varying areas (truss 2), and a space truss (truss 3).

Truss 4 is a simple, once redundant planar truss with uniform cross-sectional areas. At the load of $1.154 \times A \times \sigma_Y$ (its collapse load), the force distribution is as shown in Fig. X. From the D and γ matrices it can be seen that the assumption that member 3 is the last to yield leads to the largest values of joint displacements, and also satisfies the conditions of equations 4.17, 4.18, and 4.19.

Truss 2 (see Fig. VII as well) is found to have the compatible collapse force distribution given in Fig. XI. The last members to yield are 6 and 15, member 2 yielding first. (Note that the structure fails partially with a collapse mechanism of two degrees of freedom. The matrix u is zero so compatibility has been maintained).

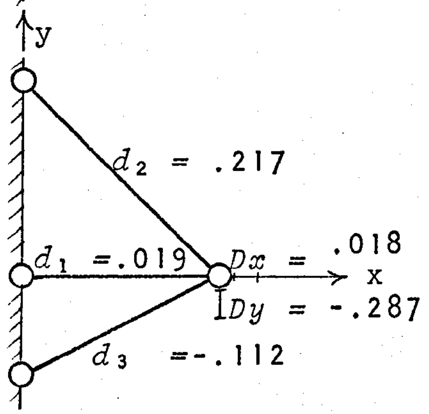
Truss 3 (see Fig. VIII also) is found to have the compatible collapse force distribution given in Fig. XII. The last member to yield is 11, member 9 yielding first. (Note that the structure fails partially with a collapse mechanism of one degree of freedom).

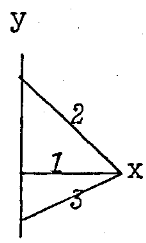
FORCES:



(Equal areas)

DISPLACEMENTS:





$p = [\ .187 \quad 1.00 \quad -.999\ ]$

If member *2* assumed last to yield,

then $\frac{1}{\varepsilon_Y} D = [\ .019 \quad -.181\ ]$

and $\gamma = [\ .187 \quad 1.000 \quad -.455\ ]$

If member *3* assumed last to yield,

then $\frac{1}{\varepsilon_Y} D = [\ .019 \quad -.287\ ]$

and $\gamma = [\ .187 \quad 1.529 \quad -.999\ ]$

Thus member *3* is last to yield.

<u>Figure X : Elastic deflexion analysis of truss 4.</u>

Truss 2. : see Figure VII.

$$p = [\ .153 \quad 1.00 \quad .625 \quad .346 \quad -.127 \quad -1.00 \quad -.563$$
$$-.307 \quad -.127 \quad .153 \quad -.125 \quad .189 \quad -.307 \quad -.153$$
$$1.00 \quad .125 \quad .307 \quad .180 \quad -.062 \quad -.346 \quad 0.000\ ]$$

Members *6* and *15* are found to yield last, giving :

$$\frac{1}{\varepsilon_Y} D = [\ .015 \quad -.399 \quad .312 \quad -.424 \quad .375 \quad -.301 \quad .166 \quad .031$$
$$.196 \quad -.282 \quad .253 \quad -.436 \quad .353 \quad -.383 \quad .365 \quad -.013$$
$$.410\ ]$$

$$\gamma = [\ .153 \quad 2.97 \quad .625 \quad .346 \quad -.127 \quad -1.00 \quad -.563$$
$$-.307 \quad -.127 \quad .153 \quad -.125 \quad .189 \quad -.307 \quad -.153$$
$$1.00 \quad .125 \quad .307 \quad .180 \quad -.062 \quad -.347 \quad 0.00\ ]$$

$$u = [\ 0.000 \quad 0.000 \quad 0.000\ ]$$

where cuts are made in members *18*, *19*, *20*.

(The load applied is the collapse load.)

<u>Figure XI : Elastic deflexion analysis of truss 2.</u>

Truss 3. : see Figure VIII.

$$p = [\ .500 \quad -.187 \quad -.040 \quad -.540 \quad .314 \quad -.000 \quad .460 \quad -.687 \quad 1.00 \quad -.540 \quad -1.00 \quad .314 \quad -.650 \quad -.443 \quad .320\ ]$$

Member *9* is found to yield last, giving :

$$\frac{1}{\varepsilon_Y} D = [\ .316 \quad -.083 \quad -.042 \quad -.412 \quad -.561 \quad -.000 \quad -.394 \quad .331 \quad .150 \quad .031 \quad -.506 \quad -.737\ ]$$

$$\gamma = [\ .500 \quad -.187 \quad -.040 \quad -.540 \quad .313 \quad -.000 \quad .460 \quad -.687 \quad 1.00 \quad -.540 \quad -3.687 \quad .313 \quad -.650 \quad -.443 \quad .320\ ]$$

$$u = [\ 0.00 \quad 0.00\ ]$$

where cuts are made in members *13, 15*

Figure XII : Elastic deflexion analysis of truss 3.

CHAPTER V. PLASTIC DESIGN

5.1 Introduction

Within a few years of the first discussions of the plastic behaviour of ductile structures, various direct and indirect methods of plastic analysis and design were described (see sections 2.1 and 2.4). Plastic theory enabled the conditions for direct design of structures to be simply stated, while retaining rationality in attempting to take into account the reserves of strength inherent in ductile, redundant structures which were not allowed for in elastic design methods.

In practice, good design attempts to reduce the cost of the materials, construction, and maintenance over the lifetime of the structure. This is not easily expressed mathematically because of the many unknown factors which may affect the cost. Plastic design for minimum weight, however, can easily be formulated mathematically, and although the minimum weight design (m.w.d.) may not really be practical, it provides an ideal for the practical designer to aim for as he takes into account the less easily quantifiable factors mentioned above.

For the description below of the direct design of minimum weight, three-dimensional, pin-jointed trusses, the assumptions are those discussed in section 2.5. They include proportional loading, neglect of changes of geometry under load, a perfectly elastic-perfectly plastic load-deformation relationship for both tension and compression, and a homogeneous material so that the weight is directly proportional to the volume. The weight per unit length of section is directly proportional to the plastic yield force of the section. The discussion assumes a continuous range of sections of uniform cross-sectional area.

5.2 The Linear Programming Problem

Following TOAKLEY (1968), the direct design of any ideal elastic-plastic three-dimensional pin-jointed truss is best formulated using a static or "safe" approach. The equilibrium is satisfied, and the yield force nowhere exceeded, W, the weight of the truss, is an upper limit (i.e., "safe" design) on W_c, the weight of the m.w.d.

For a single loading condition this can be formulated as follows. The equilibrium equation 3.6

$$Q = C\ q$$

or, with the member forces expressed in terms of the external loading system and a set of internal redundants, equation 3.13

$$q = B_o\ Q + B_1\ x \qquad ..(5.1)$$

The yield condition can be expressed as

$$\left.\begin{array}{l} \sigma_Y\ a \geq q \\ \alpha\ \sigma_Y\ a \geq -q \end{array}\right\} \qquad ..(5.2)$$

where a is the vector of member areas, $a_i = A_i$.

Then substituting equation 5.1 into equation 5.2, obtain

$$\left.\begin{array}{l} \sigma_Y\ a - B_1\ x \geq + B_o\ Q \\ \alpha\ \sigma_Y\ a + B_1\ x \geq - B_o\ Q \end{array}\right\} \qquad ..(5.3)$$

The volume or weight of the structure is obtained from

$$V = [\ell^T \vdots\ 0^T] \begin{bmatrix} a \\ \cdots \\ x \end{bmatrix} \qquad ..(5.4)$$

The linear programming problem is to minimise 5.4 subject to 5.3.

Substituting equations 3.5 and 3.22

$$Q = \sigma_Y\ A\ P$$

$$x = \sigma_Y\ A\ y$$

into equation 5.3 results in

$$\left.\begin{array}{l} a^* - B_1\ y \geq + B_o\ P \\ \alpha\ a^* + B_1\ y \geq - B_o\ P \end{array}\right\} \qquad ..(5.5)$$

where a* is the vector of area ratios, $a^*_i = A_i/A$. The linear programming problem becomes minimise $$V^* = [\ell^T \vdots 0^T] \begin{bmatrix} a^* \\ \cdots \\ y \end{bmatrix} \qquad ..(5.6)$$

subject to equation 5.5.

The reduced problem of equations 5.5 and 5.6 is formulated in terms of A, the reference area. If P has elements of modulus close to unity, then a^* and y are kept correspondingly small. The program to solve this is described in section 5.5.

5.3 Design for Several Loading Cases

The previous section describes the formulation of the design problem with a single loading case. But designing for a single loading system is hardly realistic as a structure will normally support several independent loading cases. They may act together or separately, loading or unloading independently. In elastic design and analysis this can be overcome by using superposition to obtain the worst possible situation. But the principle of superposition does not hold for plastic behaviour, and the added dangers of incremental collapse and alternating plasticity further complicate the problem.

In this section, the problem of several proportional loading cases applied alternately is studied. Toakley, in a personal communication (1969), and DORN et al. (1964) have suggested increasing the set of independent variables of the linear programming problem to include a set of redundants for each loading condition. This will ensure the m.w.d. at the expense of doubling the number of constraints with each loading case and increasing the number of independent variables. For more than a few loading cases the method becomes too large for any but the biggest computers.

The problem for c loading cases becomes

$$\text{minimise} \quad V^* = [\ell^T \;\vdots\; 0^T] \begin{bmatrix} a^* \\ \cdots \\ y_1 \\ \cdots \\ y_2 \\ \cdots \\ \vdots \\ \cdots \\ y_c \end{bmatrix} \qquad ..(5.7)$$

subject to

$$\begin{array}{lllll} a^* & - B_1 y_1 & & & \geq + B_o P_1 \\ \alpha\, a^* & + B_1 y_1 & & & \geq - B_o P_1 \\ a^* & & - B_1 y_2 & & \geq + B_o P_2 \\ \alpha\, a^* & & + B_1 y_2 & & \geq - B_o P_2 \\ \vdots & & & \ddots & \vdots \\ \alpha\, a^* & & & + B_1 y_c & \geq - B_o P_c \end{array} \qquad ..(5.8)$$

A program using this method is described in section 5.5.

A procedure for obtaining an "efficient" design which is "safe", although not the m.w.d., under many loading cases and which uses no more storage space than the single loading case problem described in section 5.2 can be formulated.

The linear programming problem becomes

$$\text{minimise} \quad V^* = [\ell^T \;\vdots\; 0^T] \begin{bmatrix} a^* \\ \cdots \\ y \end{bmatrix} \qquad ..(5.9)$$

$$\text{subject to} \quad \left.\begin{array}{l} a^* - B_1\, y \geq + R_{max} \\ \alpha\, a^* + B_1\, y \geq - R_{min} \end{array}\right\} \qquad ..(5.10)$$

where R_{maxi} is the greatest tensile load in member i due to any of the external loading cases only, obtained by

$$+R_{maxi} = \max.\{+(B_o\ P_1)_i, +(B_o\ P_2)_i, \ldots, +(B_o\ P_c)_i\} \qquad ..(5.11)$$

R_{mini} is the greatest compressive load in member i due to any of the external loading cases only, obtained by

$$-R_{mini} = \max.\{-(B_o\ P_1)_i, -(B_o\ P_2)_i, \ldots, -(B_o\ P_c)_i\} \qquad ..(5.12)$$

Thus R_{maxi} and R_{mini} define an envelope of the critical tensile and compressive member loads due to the external loading cases only.

Having obtained the optimum vector a* which minimises V^*, the collapse load factors for the designed structure under each of the loading cases can be obtained. If all are greater than unity, then the cross-sectional areas of the design can be divided by the smallest collapse load factor obtained, leading to a modified load factor of unity for this loading case. The design is then about to fail in at least one of the loading cases. This is an efficient use of material, but is not necessarily the m.w.d.: in general, it will not be, except for the single loading case when this approach is identical to that of equations 5.5 and 5.6.

A program based on equations 5.9 and 5.10 is described below.

5.4 Self-Weight Design

The design of structures using linear programming and including the self-weight of the members can be done iteratively. DORN et al. (1964) mentioned the problem, but it has not received much attention. It is merely an extension of the procedures described above.

Firstly, using any of the above procedures, obtain the optimum vector a*, assuming no self-weight, and using the actual load matrix P (for one or more loading cases). Then calculate

$$v_1^* = M\ a^* \qquad ..(5.13)$$

where v_1^* is a member volume (or weight) vector

M is the diagonal member length matrix $M_{ii} = \ell_i$

$M_{ij} = 0,\ i \neq j$

The member weight vector is obtained by multiplying by the specific weight of the material. The correction load vector P_1' is obtained by adding half the weight of each member to the vertical downwards load components acting at each end of the member.

The design procedure is repeated, using the load matrix $(P + P_1')$, to get a new design, which has a different weight from the old design. The new weight leads to a new correction matrix P_2', and the design is repeated using $(P + P_2')$ as the load matrix.

The procedure is repeated until the difference between the volumes of successive designs $(V_{n+1} - V_n)$ is sufficiently small.

In designing for many loading cases, the corrections can be applied to the tensile and compressive envelopes R_{max} and R_{min}:

$$\left.\begin{aligned} \text{new } R_{max} &= \text{old } R_{max} + B_o P_i' \\ \text{new } R_{min} &= \text{old } R_{min} + B_o P_i' \end{aligned}\right\} \quad ..(5.14)$$

A program for the design of pin-jointed trusses, allowing for self-weight, is described below.

5.5 Two Programs for Design with Several Loading Cases

In all the formulations above, (equations 5.6, 5.7, and 5.9), the coefficients of the merit functions, the member areas and structure volumes respectively, are positive or zero. These problems can most efficiently be solved using the Dual Simplex algorithm. TOAKLEY (1968) has developed such an algorithm for the m.w.d. of rigid-jointed frameworks under single loading conditions and it is this algorithm, modified slightly, which is used in one of the two programs below.

Toakley has described two means of shortening the time and reducing the storage needed in solution. One way is applicable when, as in equation 5.5, the B_o and B_1 terms cancel on addition of the equations. It leads to the explicit consideration in the tableau of only half the constraints. The other way is to use non-negative variables

z, given by

$$z = y + b \qquad ..(5.15)$$

where b are constant and chosen so that z are non-negative: b are lower limits on the estimated possible values of y.

The Dual Simplex algorithm of Toakley, employing the substitution of 5.15, has been used in a program to solve the "efficient" multi-loading case problem of equations 5.9 and 5.10, with self-weight iterations. The two-phase Standard Simplex algorithm of KRANTE and KRSIMP has been used in a program to solve the m.w.d. under multi-loading case, the equations 5.7 and 5.8.

The latter is known as MULTI-LOAD PLASTIC DESIGN (TOAKLEY) and comprises eight subroutines:

POLOAD - dimensions the arrays and calls the other subroutines in turn;

PLOPT 5 - reads the data of the structure and forms the connexion, load, length, and member incidences matrices;

RANK 2 - generates the matrices B_o and B_1, and isolates a consistent and well-conditional set of redundants;

MANYLD - sets up the problem as expressed by equations 5.7 and 5.8;

DUALP 2 - forms the dual problem and calls KRANTE and KRSIMP;

FINAL - prints the optimum member areas, values of the redundants, and the structural volume.

The former is known as SELF-WEIGHT PLASTIC DESIGN and comprises six subroutines:

SELFWT - dimensions the arrays, sets up the initial problem, and calls the other subroutines iteratively until convergence;

PLOPT 5 - reads the data of the structure and forms the connexion, load, length, and member incidences matrices;

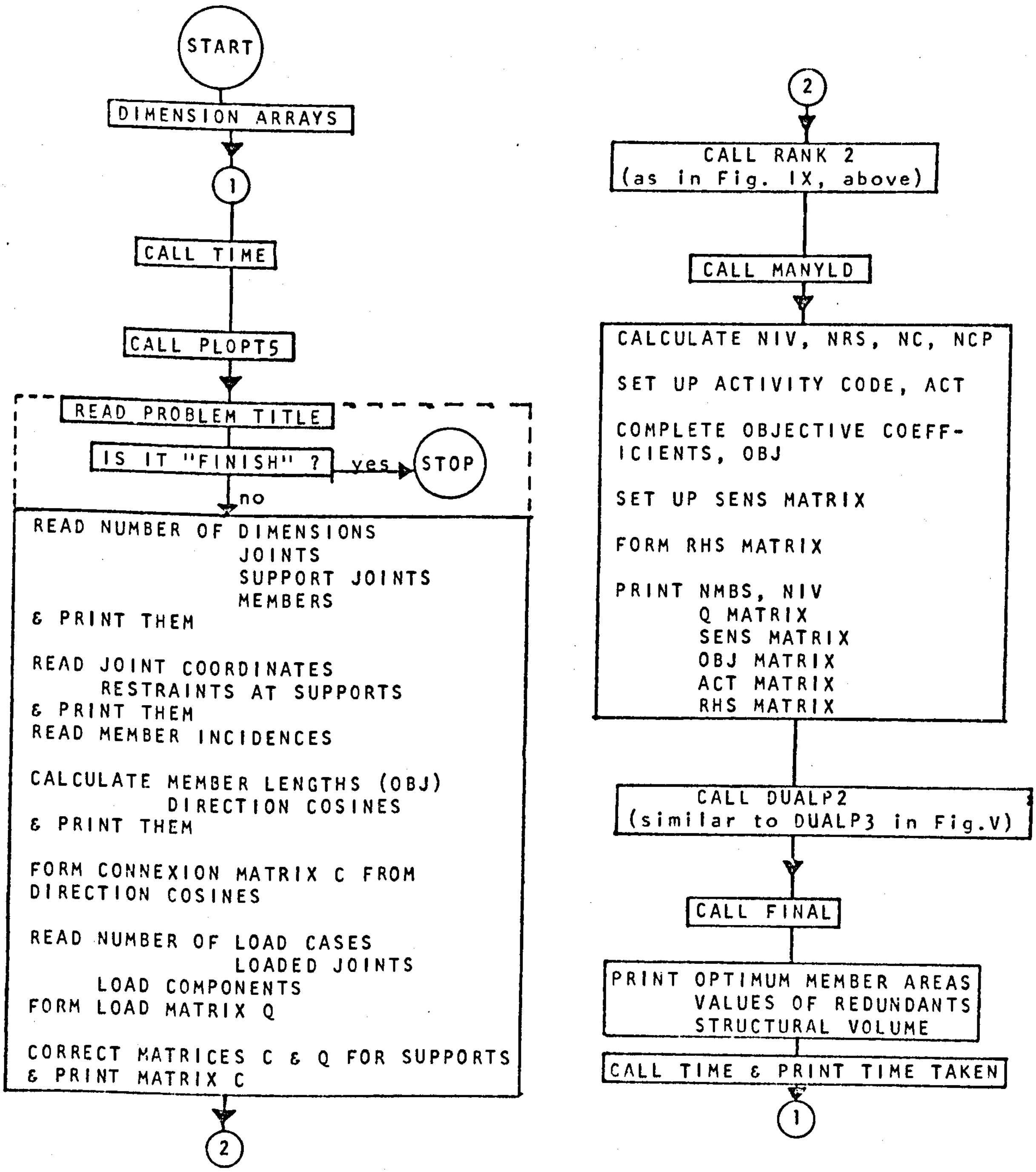


Figure XIII : Flow Diagram of MULTI-LOAD PLASTIC DESIGN

RANK 2 - generates the matrices B_o and B_1, and isolates a consistent and well-conditioned set of redundants;

SAG - adds the self-weight to the load matrix, calculates and prints the critical load envelope R_{max} and R_{min}, completes and prints the objective coefficients, and prints α, the compression coefficient, and P, the load matrix;

LOMFRE - forms a contracted linear programming tableau and solves the linear programming problem using the Dual Simplex algorithm - a modification of Toakley's LIMFRAM program;

ULTIM - prints the optimum areas and redundants sets, the minimum structural volume, and the number of iterations.

For a pin-jointed truss with m members, r of which are redundant, a comparison of the size of the linear programming problems shows:

1. for a single loading case (equations 5.5 and 5.6) using Toakley's two methods of shortening the Dual Simplex problem, number of independent variables $= m + r$

number of constraints $= m$

This particular problem is not described here.

2. for c loading cases, using the formulation of equations 5.9 and 5.10, and the Dual Simplex algorithm,

number of independent variables $= m + r$

number of constraints $= 2m$

with self-weight, this problem is iterated.

3. for c loading cases, using the formulation of equations 5.7 and 5.8, to obtain the m.w.d. using the two-phase Standard Simplex algorithm

number of independent variables $= m + 2 \times c \times r$

number of constraints $= 2 \times c \times m$

Equations 5.7 and 5.8, to obtain the m.w.d. using the Dual Simplex algorithm with Toakley's two devices:

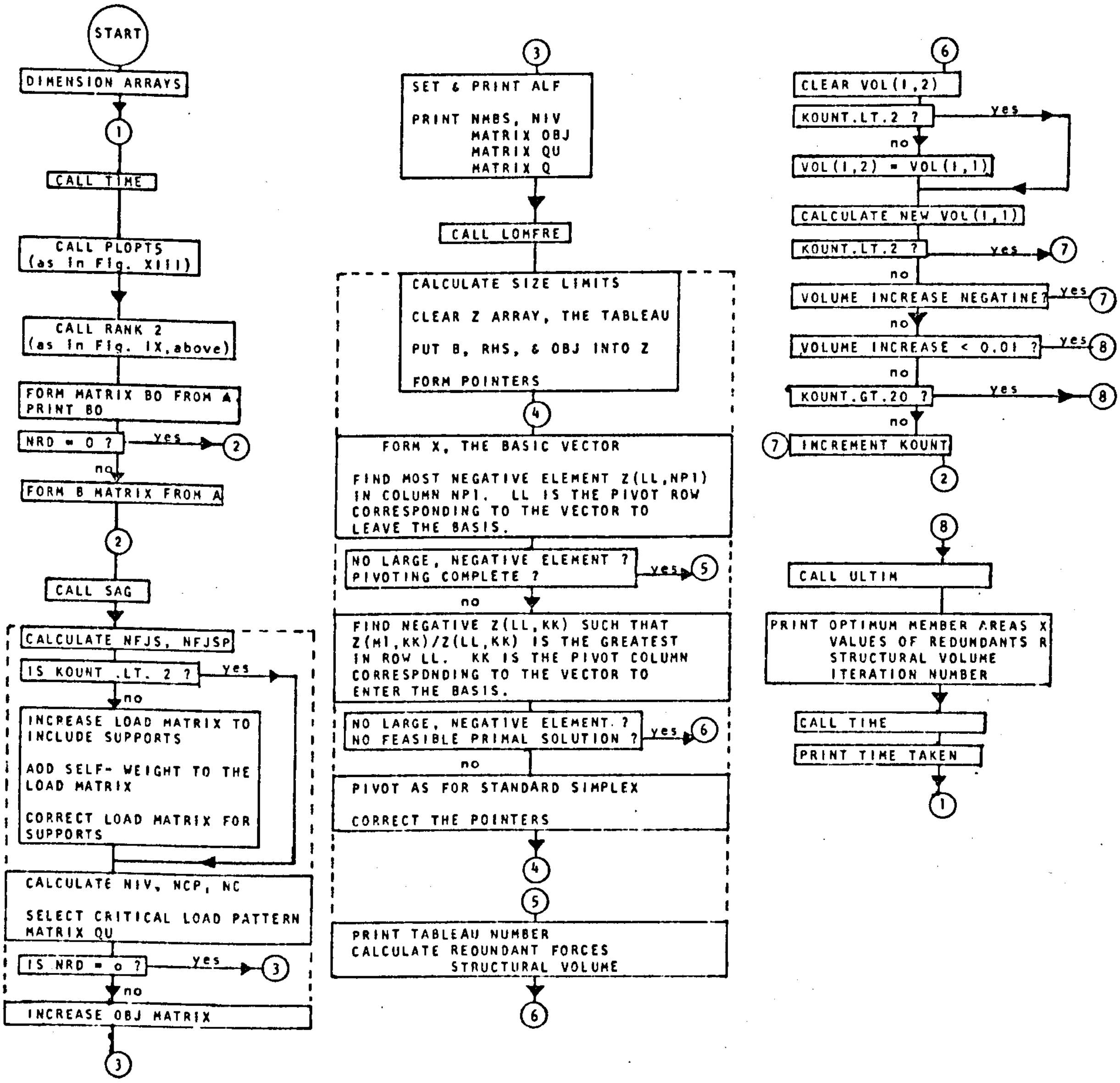


Figure XIV ; Flow Diagram of SELF-WEIGHT PLASTIC DESIGN number of independent variables $= m + c \times r$

number of constraints $= c \times m$

Programming for this method of solution of the m.w.d. is not described here.

Flow diagrams of the two programs described above are presented in Figures XIII and XIV. Full listings of the two programs are presented in Appendices C and D.

5.6 Design Examples

The designs of two trusses are compared, using each of the programs. The trusses are both fairly simple planar trusses under two loading cases: the m.w.d. program, MULTI-LOAD PLASTIC DESIGN cannot be used for large structures with several loading cases, as the problem becomes too large for the computer to handle.

Truss 5 is a three times redundant planar pin-jointed truss, as shown in Fig. XV. Loading case 1 is a unit vertical force downwards, loading case 2 is a unit horizontal force to the right. The volume of the minimum weight design (MULTI-LOAD PLASTIC DESIGN) is 30% less than the volume of the "efficient" weight design (SELF-WEIGHT PLASTIC DESIGN). After the addition of self-weight to the loading in the "efficient" case, the volume has increased by 10%.

Truss 6 is a twice redundant planar truss, as shown in Fig. XVI. Loading case 1 is a unit vertical downwards force from the bottom midspan, and loading case 2 is a unit horizontal force right, from the top midspan. The "efficient" design is over twice the volume of the m.w.d., but self-weight consideration increases it by about 10% only.

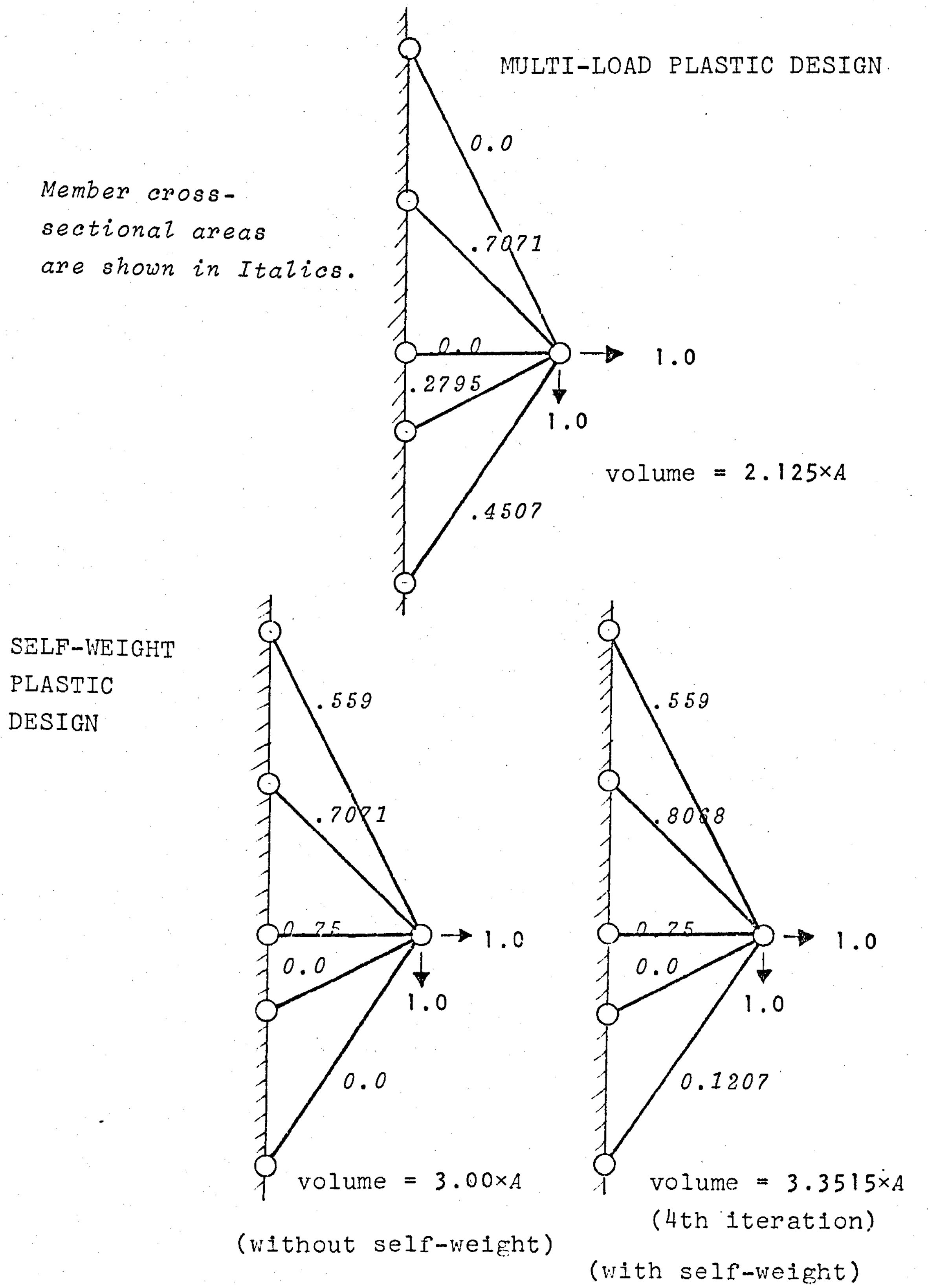


Figure XV : Plastic design of truss 5.

MULTI-LOAD
PLASTIC
DESIGN

SELF-WEIGHT
PLASTIC
DESIGN

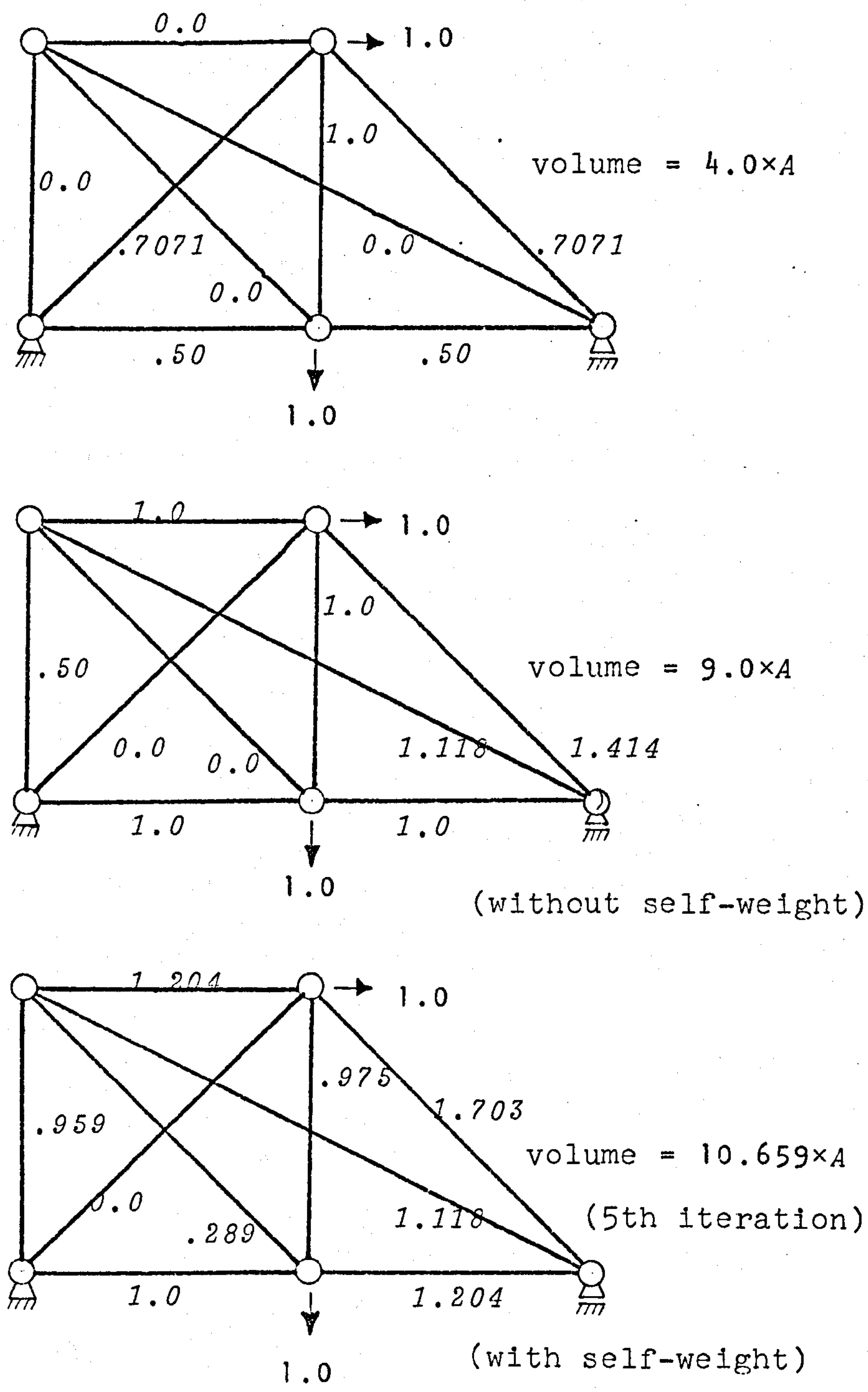


Figure XVI : Plastic design of truss 6.

CHAPTER VI. CONCLUSION

Four algorithms have been written and programmed successfully, dealing with planar or space pin-jointed trusses of varying cross-sectional areas. The four programs, respectively,

1. perform a load factor analysis;
2. perform an elastic deflexion analysis;
3. obtain the minimum weight design for one or several loading cases;
4. obtain an "efficient" weight design for one or several loading cases, taking the self-weight of the members into account.

A subroutine has been written using the Rank technique to isolate consistent and reasonably well-conditioned sets of redundants. It generates the force transformation matrices B_o and B_1, and calculates the number of degrees of freedom of any mechanism described.

An algorithm has been written to complete the constrained minimisation of the strictly convex structural complementary energy, using the piece-wise linearisation technique of REINSCHMIDT et al. (1966) with adaptive move limits. In purely elastic behaviour the minimum point is entirely unconstrained. As the loading is increased the statically admissible hyper-space shrinks as the yield constraints close in. As the first bar yields the previously unconstrained minimum becomes semi-constrained. If the structure is about to fail in complete or over-complete collapse, the feasible region is only a point at the intersection of the yield constraints. In the case of partial collapse the statically admissible region is a hyper-space with as many dimensions as the partial collapse structure has degrees of indeterminacy at collapse.

The elastic deflexions at incipient collapse have been calculated for several different types of truss. The assumption of ideal elastic-plastic load-deflexion behaviour for struts, discussed in section 2.5, is valid only if the plastic plateau required for collapse is not excessive. A strut of truss 3 (see Fig. XII) showed a compressive plastic plateau of 2.69 at collapse, (with the compression coefficient of $\alpha = 1.0$). None but the stockiest members could exhibit plastic plateaus of this size. A reduced value of α in the analysis would lead to a lower collapse load and smaller elastic deflexions at incipient collapse. The plastic plateau required would be smaller.

The collapse deflexions of structures are independent of their actual member areas: the greater the areas the stiffer the structure, but the greater the areas the stronger the structure (see equation 4.13). For a given structural and loading configuration, the collapse deflexions are dependent on the area ratios, the compression coefficient α, and the tensile yield strain ε_Y.

The minimum weight designs obtained under several loading cases have generally been much lighter than the corresponding "efficient" weight designs. Perhaps in structures of low redundancy with many separate loading cases, the "efficient" weight algorithm may offer a real advantage, but for the structures considered, the designs obtained, although "safe", were hardly "efficient" in the same sense as the m.w.d.

Designs for self-weight are about 10% greater in weight than the "efficient" designs without self-weight, assuming a specific weight of 0.10. (The designs converged after four to ten iterations).

Future Developments

Developments suggested here are of two kinds; firstly, improvements in the programming techniques, and, secondly, program modifications to allow their application to structures acting flexurally.

The Rank technique, used in all programs, could be improved by selecting as pivot in the elimination the largest remaining element in the whole of the submatrix **C**, instead of merely the largest element in each successive row. In analysis, the technique described by DENKE (1965) of taking account of the member flexibilities could give a better set of redundants (see section 3.4).

The deflexion analysis could be further automated by calculating d and n, to obtain r from equation 4.9. The ${}^{n}C_{r}$ combinations could then be calculated automatically, and hence the corresponding **D**, **d**, and γ matrices.

The program MULTI-LOAD PLASTIC DESIGN could make better use of computer storage and time by using a modified version of the subroutine LOMFRE to solve the equations 5.7 and 5.8. Both of the methods described by TOAKLEY (1968) to reduce the computer time and storage could be used (see section 5.5). (LOMFRE was twice as fast as KRANTE and KRSIMP for a case of single loading).

The most important future development, however, is the possible modification of the load factor and deflexion analysis programs for use with flexurally-acting structures. (This has already been done for design (TOAKLEY (1967), (1968)).) A simplified method for considering the bending moments only has been described by LIVESLEY (1964), but the more general case of structures, some of whose members may fail in bending, some in tension or compression, could easily be formulated.

APPENDIX A

Appendix A presents the listings of six of the eight subroutines of the program RANK PLASTICITY OPTIMISATION. This is a program to determine the collapse load of two- or three-dimensional pin-jointed trusses with ideal elastic-plastic member load-deformation behaviours in tension and compression.

The program is dimensioned:

maximum number of joints $j = 16$

" " " members $m = 49$

" " " redundant members $r = 10$

The relevant matrices of the program are:

COORD	is the matrix of joint coordinates
RELS	is the matrix of support restraints
A	is the augmented connexion matrix $[C \vdots -L]$
Z	is the Simplex tableau
SENS	is the coefficient matrix of equation 3.23
PROD	is the member force ratio vector p
AREA	is the member area ratio vector a*
MRED	is the vector of the redundant member numbers
ALF	is the compression coefficient α
NRC	is the number of degrees of freedom at the joints
NMBS	is the number of members
NRD	is the number of redundant members
NIV	is the number of independent variables

The two subroutines KRANTE and KRSIMP comprise a two-phase Standard Simplex algorithm and not shown.

```
$JOB            356879, MARKS
$TIME           3
$*              RANK PLASTICITY OPTIMISATION
$IBJOB          MAP
$IBFTC NOINV    NODECK
C PROGRAM FOR DETERMINING THE CRITICAL MEMBERS OF LOADED TRUSSES, COMPOSED
C  OF MEMBERS OF VARYING CROSS-SECTIONAL AREAS.
C  PLANAR OR SPACE TRUSSES PROCESSED
C THE PROGRAM IS COMPRISED OF EIGHT SUB-PROGRAMS -
C       NOINV  - DIMENSIONS OF VARIABLE ARRAYS
C       PLOPT3 - FORMS THE CONNEXION AND LOAD MATRICES, WITH DIFFERENT AREAS
C       RANK   - PROCESSES AND REARRANGES THE AUGMENTED MATRIX A
C       FORMS  - FORMS THE LINEAR PROGRAMMING PROBLEM
C       DUALP  - FORMS THE Z ARRAY
C       KRANTE + KRSIMP - THE L. P. SUB-PROGRAMS
C       ANSWER - OUTPUTS THE RESULTS
C
      REAL COORD(16,3),RELS(16,3),A(50,50),Z( 60,150),SENS(50,10),
     * PROD(50),CHS(10),AREA(50)
      INTEGER KUT(150),IDEG(150),NAME(150),MARK( 60),MRED(10),
     * MOP(50),KNUT(10)
C
      DO 170 LOOK = 1,20
      CALL TIME
      CALL PLOPT3 (COORD,RELS,A,AREA,16,50,NRC,NMBS)
      CALL RANK (A,MRED,KNUT,MOP,50,10,NRC,NMBS,NRD)
      CALL FORMS (A,SENS,AREA,10,50,NMBS,NRD,NIV)
      CALL DUALP3 (SENS,Z,1,CHS,KUT,IDEG,NAME,MARK,50,10, 60,150,NMBS,
     * NIV,1.)
      CALL ANSWER (SENS,Z,PROD,50, 60,150,NIV,NMBS,NRD)
      CALL TIME (NM,NS,NSS)
C
      WRITE (6,169) NM, NS, NSS
  169 FORMAT(1H-,32HTIME TAKEN FOR ABOVE STRUCTURE -/20X,I4,5H MINS,I6,
     * 5H SECS,I6,6H SSECS)
  170 CONTINUE
      CALL EXIT
      END



$IBFTC PLOPT3
      SUBROUTINE PLOPT3 (COORD,RELS,A,AREA,NTJS,NTMBS,NRC,NMBS)
      REAL COORD(NTJS,3),RELS(NTJS,3),A(NTMBS,NTMBS),TYPE(2),XYZ(3),
     * PRNAME(13),DRNCOS(3),QJ(3),LENGTH,AREA(NTMBS)
      DATA TYPE/6H PLANE, 6H SPACE/, XYZ/1HX, 1HY, 1HZ/, FIN/6HFINISH/
C
C READ HEADER CARD
      READ (5,10) PRNAME
   10 FORMAT (13A6)
      IF (PRNAME(1).EQ.FIN) CALL EXIT
      WRITE (6,15) PRNAME
   15 FORMAT (1H1, 13A6)
C
C READ PROBLEM PARAMETERS -
```

```
C   JF = 2 FOR TWO-DIMENSIONAL TRUSS
C   JF = 3 FOR THREE-DIMENSIONAL TRUSS
C  NJS = NUMBER OF JOINTS IN TRUSS
C NSJS = NUMBER OF SUPPORT JOINTS IN TRUSS
C NMBS = NUMBER OF MEMBERS IN TRUSS
      WRITE (6,20)
   20 FORMAT (1H0, 10X, 6HDATA -)
      READ(5,25) JF, NJS, NSJS, NMBS
   25 FORMAT (20I4)
      NFJS = NJS - NSJS
C
C OUTPUT PROBLEM PARAMETERS
      WRITE (6,30) TYPE(JF - 1)
   30 FORMAT (1H0,20X,20HTYPE OF STRUCTURE  -,A6,29H TRUSS WITH THE CROSS-S
     *S-SECTION/42X,27HAREA OF EACH MEMBER VARYING)
      WRITE (6,35) NJS, NSJS, NMBS
   35 FORMAT (1H0,20X,20HNUMBER OF JOINTS    =,I4/31X,10HSUPPORTS =,I4/3
     *1X,10HMEMBERS  =,I4)
C
C READ JOINT NUMBERS AND COORDINATES -
C FREE JOINTS MUST BE NUMBERED FIRST, THEN SUPPORTS
C IF RELS(N,1) = 1. ... RESTRAINT AT N IN X-DIRN.
C IF RELS(N,2) = 1. ... RESTAINT AT N IN Y-DIRN.
C IF RELS(N,3) = 1. ... RESTRAINT AT N IN Z-DIRN.
      DO 39 I = 1,NJS
      DO 39 J = 1,JF
   39 RELS(I,J) = 0.
      DO 40 I = 1,NJS
      READ (5,45) N, (COORD(N,J),J = 1,JF)
   45 FORMAT (I4, 3F8.4)
      IF (N.LE.NFJS) GO TO 40
      READ (5,46) (RELS(N,J),J = 1,JF)
   46 FORMAT (3F4.1)
   40 CONTINUE
C
C OUTPUT JOINT NUMBERS AND COORDINATES - FIRST FOR FREE JOINT AND THEN
C SUPPORT JOINTS
      WRITE (6,50) (XYZ(I),I=1,JF)
   50 FORMAT (1H0,20X,20HJOINT COORDINATES  -//28X,5HJOINT,10X,3(A1,11X))
     *)
      WRITE (6,55)
   55 FORMAT (23X,4HFREE)
      DO 60 I = 1,NJS
      WRITE (6,65) I, (COORD(I,J),J=1,JF)
   60 IF (I.EQ.NFJS) WRITE (6,70)
   65 FORMAT(1H ,I31,2X,6F12.4)
   70 FORMAT (23X,7HSUPPORT)
      NFJSP = NFJS + 1
      WRITE (6,47) ((RELS(I,J),J = 1,JF),I = NFJSP,NJS)
   47 FORMAT (1H0,20X,6HRELS -/27X,(6F5.1))
      WRITE (6,75) (XYZ(I),I = 1,JF)
   75 FORMAT (1H0,20X,17HMEMBER DETAILS  -//31X,6HMEMBER,6X,5HSTART,7X,
     * 3HEND,5X,6HLENGTH,6X,4HAREA,2X,3(3X,7HDRNCOS-,A1))
C
C LOOP ENTERED FOR ALL MEMBERS
      NRC = JF*NJS
      DO 79 I = 1,NRC
      DO 79 J = 1,NMBS
   79 A(I,J) = 0.
      NC = NMBS + 1
```

```
C
C
C READ MEMBER INCIDENCES AND PROPERTIES
      DO 120 MB = 1,NMBS
      READ (5,80) M,JNTST,JNTEND,AREAM
   80 FORMAT (3I4, F12.4)
      IF (                          (JNTEND.LE.NJS).AND.(JNTST.LT.JNTEND)) GO
     * TO 90
      WRITE (6,85) M
   85 FORMAT (1H-,20H INCIDENCE OF MEMBER,I4,39H INCORRECTLY SPECIFIED.
     * JOB TERMINATED)
      CALL EXIT
   90 CONTINUE
      IF (AREAM.EQ.0.) AREAM = 1.
      AREA(M) = AREAM
C CALCULATE MEMBER LENGTHS
      TLGTH = 0.
      DO 95 I = 1,JF
   95 TLGTH = TLGTH + (COORD(JNTEND,I) - COORD(JNTST,I))**2
      LENGTH = SQRT(TLGTH)
C
C CALCULATE MEMBER DIRECTION COSINES
      DO 100 I = 1,JF
  100 DRNCOS(I) = (COORD(JNTEND,I) - COORD(JNTST,I))/LENGTH
C
C OUTPUT MEMBER INCIDENCES AND PROPERTIES
      WRITE (6,105) M,JNTST,JNTEND,LENGTH,AREAM,(DRNCOS(I),I = 1,JF)
  105 FORMAT (1H , 23X, 3I11,2F12.4, 6E11.3)
C
C ADD MEMBER DIRECTION COSINES INTO THE CONNEXION MATRIX
      LROCS = JF*JNTST
      IROCS = LROCS - JF + 1
      DO 110 I = IROCS, LROCS
      J = I - IROCS + 1
  110 A(I,M) = -DRNCOS(J)
      LROCE = JF*JNTEND
      IROCE = LROCE - JF +1
      DO 115 I = IROCE, LROCE
      J = I - IROCE + 1
  115 A(I,M) = DRNCOS(J)
  120 CONTINUE
C
C
C READ SIZES OF LOAD COMPONENT AT EACH JOINT
C MUST NUMBER IN ORDER X, Y, Z, AT EACH JOINT
C NLJTS = NUMBER OF LOADED JOINTS
      READ (5,140) NLJTS
  140 FORMAT (16I4)
      DO 145 I = 1,NRC
  145 A(I,NC) = 0.
      DO 155 NO = 1,NLJTS
      READ (5,150) N, (QJ(I), I = 1,JF)
  150 FORMAT (I4,9F8.4)
      DO 155 I = 1,JF
      J = JF*(N-1) + I
  155 A(J,NC) = QJ(I)
C
C TO CORRECT MATRICES C AND Q FOR SUPPORTS
      NFJSP = NFJS + 1
      M = 0
```

```
      DO 157 I = NFJSP, NJS
      DO 157 J = 1, JF
      IF (RELS(I,J).NE.1.) GO TO 157
      K = JF*(I - 1) + J - M
      NRC = NRC - 1
      DO 156 L = K, NRC
      N = L + 1
      A(L,NC) = A(N,NC)
      DO 156 II = 1, NMBS
      A(L,II) = A(N,II)
  156 CONTINUE
      M = M + 1
  157 CONTINUE
C
C OUTPUT CONNEXION MATRIX
      WRITE (6,125)
  125 FORMAT ( 1H-, 18HCONNEXION MATRIX C/)
      DO 130 I = 1,NRC
  130 WRITE (6,135) (A(I,J),J = 1,NMBS)
  135 FORMAT (1H , 10E12.3)
C
C OUTPUT THE LOAD MATRIX
      WRITE (6,160)
  160 FORMAT (1H-,13HLOAD MATRIX Q/)
      WRITE (6,165) (A(I,NC),I = 1,NRC)
  165 FORMAT (1H ,(20F6.3))
  170 CONTINUE
      RETURN
      END



$IBFTC RANK
      SUBROUTINE RANK (A,MRED,KNUT,MOP,NTMBS,NTIV,NR,NCC,NRD)
      REAL A(NTMBS,NTMBS)
      INTEGER MRED(NTIV),KNUT(NTIV),MOP(NTMBS)
C
C SET UP UNIT SUBMATRIX
      NC = NCC + 1
      DO 4 I = 1,NR
    4 A(I,NC) = -A(I,NC)
C
C JORDAN ELIMINATION
      DO 25 I = 1,NR
      DEN = 0.
      DO 5 J = 1,NCC
      AB = ABS(A(I,J))
      DN = ABS(DEN)
      IF (AB.GT.DN) JJ = J
      IF (AB.GT.DN) DEN = A(I,J)
    5 CONTINUE
      IF (DEN.EQ.0.) GO TO 25
      DO 10 K = 1,NC
   10 A(I,K) = A(I,K)/DEN
      DO 20 L = 1,NR
      IF (L.EQ.I) GO TO 20
      FAC = A(L,JJ)
      DO 15 M = 1,NC
   15 A(L,M) = A(L,M) - A(I,M)*FAC
```

```
   20 CONTINUE
   25 CONTINUE
  301 FORMAT (1H /(20F6.2))
C
C CHECK FOR INCONSISTENCY AND DEPENDENCE
      NDF = 0
      DO 65 I = 1,NR
      J = 0
   45 J = J + 1
      IF (J.GT.NC) GO TO 55
      IF (A(I,J).EQ.0.) GO TO 45
      IF (J.LE.NCC) GO TO 65
      NDF = NDF + 1
      GO TO 65
   55 NR = NR - 1
      DO 60 K = I,NR
      L = K + 1
      DO 60 M = 1,NC
   60 A(K,M) = A(L,M)
   65 CONTINUE
      IF (NDF.EQ.0) GO TO 66
      WRITE (6,50) NDF
   50 FORMAT (1H-,28HTHE STRUCTURE IS A MECHANISM/30HNUMBER OF DEGREES OF
     *F FREEDOM =,I3)
   66 CONTINUE
C
C ISOLATE REDUNDANCIES
      NRD = 0
      DO 75 J = 1,NCC
      KOUNT = 0
      DO 70 I = 1,NR
   70 IF (A(I,J).NE.0.) KOUNT = KOUNT + 1
      IF (KOUNT.LE.1) GO TO 75
      NRD = NRD + 1
      MRED(NRD) = J
   75 CONTINUE
C
C REDUNDANCIES ... CHECK, OUTPUT
      I = NCC - NR
      IF (I.NE.NRD) WRITE (6,80)
   80 FORMAT (1H-,22H(COLS - ROWS) .NE. NRD)
      IF (NRD) 85,85,95
   85 WRITE (6,90)
   90 FORMAT (1H0,32HSTRUCTURE IS ALREADY DETERMINATE)
      GO TO 105
   95 WRITE (6,100) NRD,(MRED(I),I = 1,NRD)
  100 FORMAT (1H0,13HSTRUCTURE IS ,I2,16H TIMES REDUNDANT/
     * 22H REDUNDANT MEMBERS ARE,40I3)
  105 CONTINUE
C
C MOVE ALL REDUNDANT COLUMNS TO RHS OF MATRIX A
      IF (NRD.LE.0) GO TO 135
      DO 110 K = 1,NRD
  110 KNUT(K) = 0
      DO 130 I = 1,NRD
      IF (MRED(I).GT.NR) GO TO 130
      J = NR
  115 J = J + 1
      K = 0
  120 K = K + 1
```

```
      IF (MRED(K).EQ.J) GO TO 115
      IF (K.LT.NRD) GO TO 120
      K = MRED(I)
      DO 125 L = 1,NR
      BLOC = A(L,J)
      A(L,J) = A(L,K)
      A(L,K) = BLOC
  125 CONTINUE
      KNUT(I) = MRED(I)
      MRED(I)= J
  130 CONTINUE
  135 CONTINUE
C
C FORM UNIT MATRIX IN A(NR,NR)
      DO 155 LOO = 1,2
      DO 140 I = 1,NR
      DO 140 J = 1,NR
      IF (A(I,J).EQ.0.) GO TO 140
      MOP(I) = J
  140 CONTINUE
      DO 155 I = 1,NR
      IF (MOP(I).EQ.I) GO TO 155
      DO 145 K = I,NR
      IF (MOP(K).EQ.I) L=K
  145 CONTINUE
      DO 150 J = 1,NC
      BLOC = A(I,J)
      A(I,J) = A(L,J)
      A(L,J) = BLOC
  150 CONTINUE
      MOP(L) = MOP(I)
  155 CONTINUE
C
C INCREASE A TO GET COMPLETE INTERNAL LOAD SYSTEM
      K = NR + 1
      IF (NRD.LE.0) GO TO 170
      DO 165 I = K,NCC
      DO 160 J = K,NC
  160 A(I,J) = 0.
  165 A(I,I) = -1.
  170 CONTINUE
C
C BACK TO ORIGINAL ORDER OF ELEMENT FORCES
      IF (NRD.EQ.0) GO TO 174
      DO 173 I = 1,NRD
      IF (KNUT(I).EQ.0) GO TO 173
      J = KNUT(I)
      M = MRED(I)
      MRED(I) = J
      DO 172 L = K,NC
      BLOC = A(J,L)
      A(J,L) = A(M,L)
      A(M,L) = BLOC
  172 CONTINUE
  173 CONTINUE
  174 CONTINUE
      RETURN
      END
```

```
$IBFTC FORMS   NODECK
      SUBROUTINE FORMS (A,SENS,AREA,NTIV,NTMBS,NMBS,NRD,NIV)
      REAL A(NTMBS,NTMBS),SENS(NTMBS,NTIV),AREA(NTMBS)
C
C FORM ACT, CON, OPT, X(INITIAL), LHS,RHS,OBJ
      NIV = NRD + 1
C
C FORM SENSITIVITIES
      DO 20 I = 1,NIV
      K = NMBS + 2 - I
      DO 20 J = 1,NMBS
   20 SENS(J,I) = -A(J,K)/AREA(J)
      IF (NIV.EQ.1) GO TO 30
      DO 25 I = 2,NIV
      DO 25 K = 1,NMBS
   25 SENS(K,I) = -SENS(K,I)
   30 CONTINUE
C OUTPUT THE OPTIMISATION SUBROUTINE ARGUMENTS
      WRITE (6,601)NMBS,NIV
  601 FORMAT (1H , 5HNMBS-,I4/6H NIV- ,I4)
      WRITE(6,602)((SENS(I,J),I=1,NMBS),J=1,NIV )
  602 FORMAT(1H0,6HSENS -/(10E12.3))
      RETURN
      END


$IBFTC DUALP3
      SUBROUTINE DUALP3 (SENS,Z,IPRNT,CHS,KUT,IDEG,NAME,MARK,NTRS,
     * NTMBS,NRZ,NCZ,M,N,ALF)
C
C           LINEAR PROGRAMMING BY SOLUTION OF DUAL
C
C           OUTPUT INDEX 'IPRNT' FOR THIS SUBROUTINE
C            IPRNT = 0  NO OUTPUT
C            IPRNT = 1  OUTPUT IFAS,MOVE LIMITS
C            IPRNT = 2  OUTPUT IFAS,MOVE LIMITS,Z-ARRAYS ETC.
C            IPRNT = 3  AS FOR 2 PLUS D-ROW OUTPUT FROM KRSIMP
C
      REAL SENS(NTRS,NTMBS),Z(NRZ,NCZ),CHS(NTMBS)
      INTEGER        OPT1,KUT(NCZ),IDEG(NCZ),NAME(NCZ),MARK(NRZ)
C
C           CLEAR Z-ARRAY
      DO 100 I = 1,NRZ
      DO 100 J = 1,NCZ
  100 Z(I,J) = 0.
C           SET DUAL SENSITIVITIES, OBJECTIVE COEFFICIENTS
C           AND ACTIVITY CODES IN Z-ARRAY
      ND = 0
      DO 113 J = 1,M
  110 ND = ND + 1
       DO 111 I = 1,N
      Z(I,J) = SENS(J,I)
      KK = M + ND
  111 Z(I,KK) = -SENS(J,I)
      Z(N + 1,J) = -1.
      Z(N + 1,KK) = -ALF
```

```
  113 CONTINUE
C           SET DUAL L.H. AND R.H. SIDES AND
C           CONSTRAINT CODES IN Z-ARRAY
  121 NZ = M + ND
  123 DO 130 I = 1,N
      Z(I,NZ + 3) = 0.
  125 Z(I,NZ+1) = 3.
  130 CONTINUE
      Z(1,NZ + 3) = 1.
C           MAKE R.H.SIDES OF DUAL CONSTRAINTS POSITIVE
      DO 135 I = 1,N
      CHS(I) = -1.
  135 CONTINUE
C           OUTPUT Z-ARRAY BEFORE CALLING KRANTE
      MZ = N
      MZ2 = N + 2
      NZ3 = NZ + 3
      IF (IPRNT.LT.2) GO TO 140
      WRITE (6,136)
  136 FORMAT(30H0Z-ARRAY BEFORE CALLING KRANTE/)
      DO 137 I = 1,MZ2
  137 WRITE (6,138) (Z(I,J),J=1,NZ3)
  138 FORMAT(1H ,10E12.3)
  140 CONTINUE
      CALL KRANTE (Z,MZ,NZ,IFAS,KUT,NRZ,NCZ)
C           OUTPUT Z-ARRAY ETC. BEFORE CALLING KRSIMP
      MZ2 = MZ + 2
      NZ1 = NZ + 1
      IF (IPRNT.LT.2) GO TO 145
      WRITE (6,141)
  141 FORMAT(30H0Z-ARRAY BEFORE CALLING KRSIMP/)
      DO 142 I = 1,MZ2
  142 WRITE (6,138) (Z(I,J),J=1,NZ1)
      WRITE (6,144) IFAS,(KUT(J),J=1,NZ)
  144 FORMAT(7H0IFAS =,I4/12H0KUT ARRAY -/(1H ,24I5))
  145 CONTINUE
      CALL KRSIMP (Z,IFAS,MZ,NZ,NAME,IDEG,KUT,NRZ,NCZ,IPRNT,MARK)
      IF (IFAS.EQ.3) GO TO 162
C           OUTPUT Z-ARRAY ON RETURN FROM KRSIMP
      MZ1 = MZ + 1
      IF (IPRNT.LT.2) GO TO 150
      WRITE (6,146)
  146 FORMAT (30H0Z-ARRAY ON RETURN FROM KRSIMP/)
      DO 147 I = 1,MZ1
  147 WRITE (6,138) (Z(I,J),J=1,NZ1)
  150 CONTINUE
      IF (IPRNT.LT.1) GO TO 152
      WRITE (6,151) IFAS
  151 FORMAT (7H0IFAS =,I4)
  152 CONTINUE
C           SET VALUES OF PRIMAL VARIABLES IN FIRST ROW
C           OF Z-ARRAY, PRIMAL OBJECTIVE IN Z(2,1)
      DO 154 J = 1,MZ
  154 Z(1,J) = CHS(J) * Z(MZ1,J)
      Z(2,1) = -Z(2,1)
  160 CONTINUE
      RETURN
  162 Z(2,1) = 0.0
      RETURN
      END
```

```
$IBFTC ANSWER  NODECK
      SUBROUTINE ANSWER (SENS,Z,PROD,NTMBS,NRZ,NCZ,NIV,NMBS,NRD)
      REAL SENS(NTMBS,NTMBS), Z(NRZ,NCZ), PROD(NTMBS)
C OUT PUT OPTIMUMM VALUES OF LAMDA
      WRITE (6,10)
   10 FORMAT (1H-,50H**************************************************)
      WRITE (6,15) Z(1,1)
   15 FORMAT (1H-,15HOPTIMUM LAMDA -,E16.8)
C
C OUTPUT MEMBER FORCES
      DO 20 I = 1,NMBS
      PROD(I) = 0.
      DO 20 J = 1,NIV
   20 PROD(I) = PROD(I) + SENS(I,J)*Z(1,J)
      WRITE (6,30) (PROD(I),I = 1,NMBS)
   30 FORMAT (1H0, 16HPRODUCT MATRIX -/(6E12.4))
      WRITE (6,50) (Z(1,I),I = 1,NIV)
   50 FORMAT (1H0, 7HZ ROW -/(10E12.4))
      RETURN
      END
```

APPENDIX B

Appendix B presents the listings of eight of the ten subroutines of the program MINIMUM COMPLEMENTARY ENERGY. This is a program to determine the elastic deflexions of two- or three-dimensional pin-jointed trusses, with ideal elastic-plastic member load-deformation behaviours, at any loading up to collapse load. Members, having yielded, are assumed not to unload.

The program is dimensioned:

maximum number of joints $j = 16$
maximum number of members $m = 50$
maximum number of redundant members $r = 10$

The relevant matrices of the program are:

COORD is the matrix of joint coordinates
RELS is the matrix of support restraints
A is the augmented matrix $[C \vdots I]$
Q is the joint load vector L
LEN is the member length vector ℓ
AREA is the member area ratio vector a^*
SENS is the coefficient matrix G_1
OBJ is the vector of objective coefficients (equation 4.21)
LHS is the vector of the left hand side constraints (equation 4.21)
RHS " the vector of the right hand side constraints (equation 4.21)
X is the vector of positive redundant force ratios s
Z is the Simplex tableau
QL is the member load vector g
PROD is the member force ratio vector p
D is the joint displacement vector D, and also u
DLGTH is the member deformation vector d, and also γ
C is the connexion matrix C
MRED is the vector of the redundant member numbers
NYLD is the vector of the yielded member numbers
ALF is the compression coefficient α

The two subroutines KRANTE and KRSIMP describe a two-phase Standard Simplex algorithm and are not listed.

```
$JOB            357531, MARKS
$TIME           3
$*              MINIMUM COMPLEMENTARY ENERGY
$IBJOB          MAP
$IBFTC ENERGY   NODECK
C
C PROGRAM WHICH, USING THE HAAR-VON KARMAN PRINCIPLE, MINIMISES THE STRAIN
C   ENERGY OF REDUNDANT STRUCTURES TO OBTAIN THE CORRECT SETS OF FORCES UNDE
C  ELASTIC - PERFECTLY PLASTIC BEHAVIOUR.  EQUILIBRIUM AND YIELD CRITERIA
C   ARE SATISFIED.
C
      REAL COORD(16,3),RELS(16,3),A(50,100),Q(50),LEN(50),AREA(50),
     * SENS(50,10),OBJ(10),LHS(50),RHS(50),X(10),Z( 60,150),QL(50),
     * CHS(10),PROD(50),D(50),DLGTH(50),C(50,50),BOX(10)
      INTEGER MRED(10),KNUT(10),MOP(50),KUT(150),IDEG(150),NAME(150),
     * MARK (60),NYLD(50),ADAP(10,2)
      EQUIVALENCE (Z,A)
      ISW1 = 2
      ISW2 = 2
      ALF = 1.
C
      DO 170 LOOK = 1,20
      CALL TIME
      CALL PLOPT7 (COORD,RELS,A,Q,LEN,AREA,16,50,100,NRC,NMBS)
      DO 5 I = 1,NRC
      DO 5 J = 1,NMBS
    5 C(I,J) = A(I,J)
      CALL RANK 2 (A,MRED,KNUT,MOP,50,100,10,NRC,NMBS,NRD)
      IF (NRD.EQ.0) GO TO 30
C        IF ISW1 = 1 READ LOAD FACTORS
C        IF ISW2 = 1 READ NRD,NY, (NYLD(I),I = 1,NY)
C        IF ISW2 = 2 SEARCH FOR NY AMONG MEMBERS
C        IF ISW2 = 3 NO ELIMANATION OF MEMBERS - NY = NRD = 0
      ISW1 = 1
      ISW2 = 1
      CALL HAAR (Q,AREA,A,QL,SENS,X,MRED,50,100,10,NMBS,NRD,NRC,ALF,ISW1)
     * )
      KOUNT = 0
   10 KOUNT = KOUNT + 1
      CALL KARMAN (LEN,AREA,QL,SENS,OBJ,X,LHS,RHS,50,10,NRD,NMBS,ALF)
      CALL DUALP5 (SENS,OBJ,LHS,RHS,X,Z,1,CHS,KUT,IDEG,NAME,MARK,BOX,
     * ADAP,50,10, 60,150,NMBS,NRD,3,100.,KOUNT)
      IF (KOUNT.LE.NRD) GO TO 16
      DO 15 I = 1,NRD
      IF (ABS(Z(1,I)).LT.1.E-06) Z(1,I) = 0.
   15 X(I) = X(I) + Z(1,I)
      GO TO 20
   16 DO 17 I = 1,NRD
      IF (ABS(Z(1,I)).LT.1.E-06) Z(1,I) = 0.
   17 X(I) = X(I) + Z(1,I)*.9
   20 CONTINUE
      NET = 0
      DO 21 I = 1,NRD
      IF (BOX(I).LE.1.E-05) GO TO 25
   21 IF (Z(1,I).EQ.0.) NET = NET + 1
      IF (NET.EQ.NRD) GO TO 25
      IF (ABS(Z(2,1)).GT.1.E-08) GO TO 10
   25 CALL NOWEND (Z,QL,SENS,PROD,LEN,AREA,X,50,60,150,10,NMBS,NRD,KOUNT)
     * )
      CALL DEFLN (A,LEN,PROD,NYLD,D,DLGTH,C,MRED,KNUT,MOP,50,10,100,
```

```
              :D,NRC,ISW2,ALF)
              !E (NM,NS,NSS)
C
              ),169) NM, NS, NSS
      FORMAT(1H-,32HTIME TAKEN FOR ABOVE STRUCTURE -/2 X,I4,5H MINS,I6,
     * 5H SECS,I6,6H SSECS)
  170 CONTINUE
      CALL EXIT
      END
```

```
$IBFTC PLOPT7
      SUBROUTINE PLOPT7 (COORD,RELS,A,Q,OBJ,AREA,NTJS,NTMBS,NTCA,
     * NRC,NMBS)
      REAL COORD(NTJS,3),RELS(NTJS,3),A(NTMBS,NTCA),TYPE(2),XYZ(3),
     * PRNAME(13),DRNCOS(3),QJ(3),LENGTH,Q(NTMBS),AREA(NTMBS)
      REAL OBJ(NTMBS)
      DATA TYPE/6H PLANE, 6H SPACE/, XYZ/1HX, 1HY, 1HZ/, FIN/6HFINISH/
C
C READ HEADER CARD
      READ (5,10) PRNAME
   10 FORMAT (13A6)
      IF (PRNAME(1).EQ.FIN) CALL EXIT
      WRITE (6,15) PRNAME
   15 FORMAT (1H1, 13A6)
C
C READ PROBLEM PARAMETERS -
C   JF = 2 FOR TWO-DIMENSIONAL TRUSS
C   JF = 3 FOR THREE-DIMENSIONAL TRUSS
C  NJS = NUMBER OF JOINTS IN TRUSS
C NSJS = NUMBER OF SUPPORT JOINTS IN TRUSS
C NMBS = NUMBER OF MEMBERS IN TRUSS
      WRITE (6,20)
   20 FORMAT (1H0, 10X, 6HDATA -)
      READ(5,25) JF, NJS, NSJS, NMBS
   25 FORMAT (20I4)
      NFJS = NJS - NSJS
C
C OUTPUT PROBLEM PARAMETERS
      WRITE (6,30) TYPE(JF - 1)
   30 FORMAT (1H0,20X,20HTYPE OF STRUCTURE  -,A6,27H TRUSS WITH THE CROSS-S
     *S-SECTION/42X,27HAREA OF EACH MEMBER VARYING)
      WRITE (6,35) NJS, NSJS, NMBS
   35 FORMAT (1H0,20X,20HNUMBER OF JOINTS   =,I4/31X,10HSUPPORTS =,I4/3
     *1X,10HMEMBERS  =,I4)
C
C READ JOINT NUMBERS AND COORDINATES -
C FREE JOINTS MUST BE NUMBERED FIRST, THEN SUPPORTS
C IF RELS(N,1) = 1. ... RESTRAINT AT N IN X-DIRN.
C IF RELS(N,2) = 1. ... RESTAINT AT N IN Y-DIRN.
C IF RELS(N,3) = 1. ... RESTRAINT AT N IN Z-DIRN.
      DO 39 I = 1,NJS
      DO 39 J = 1,JF
   39 RELS(I,J) = 0.
      DO 40 I = 1,NJS
      READ (5,45) N, (COORD(N,J),J = 1,JF)
   45 FORMAT (I4, 3F8.4)
      IF (N.LE.NFJS) GO TO 40
```

```
      READ (5,46) (RELS(N,J),J = 1,JF)
   46 FORMAT (3F4.1)
   40 CONTINUE
C
C OUTPUT JOINT NUMBERS AND COORDINATES - FIRST FOR FREE JOINT AND THEN
C SUPPORT JOINTS
      WRITE (6,50) (XYZ(I),I=1,JF)
   50 FORMAT (1H0,20X,20HJOINT COORDINATES  -//28X,5HJOINT,10X,3(A1,11X))
     *)
      WRITE (6,55)
   55 FORMAT (23X,4HFREE)
      DO 60 I = 1,NJS
      WRITE (6,65) I, (COORD(I,J),J=1,JF)
   60 IF (I.EQ.NFJS) WRITE (6,70)
   65 FORMAT(1H ,I31,2X,6F12.4)
   70 FORMAT (23X,7HSUPPORT)
      NFJSP = NFJS + 1
      WRITE (6,47) ((RELS(I,J),J = 1,JF),I = NFJSP,NJS)
   47 FORMAT (1H0,20X,6HRELS -/27X,(6F5.1))
      WRITE (6,75) (XYZ(I),I = 1,JF)
   75 FORMAT (1H0,20X,17HMEMBER DETAILS  -//31X,6HMEMBER,6X,5HSTART,7X,
     * 3HEND,5X,6HLENGTH,6X,4HAREA,2X,3(3X,7HDRNCOS-,A1))
C
C LOOP ENTERED FOR ALL MEMBERS
      NRC = JF*NJS
      DO 79 I = 1,NRC
      DO 79 J = 1,NMBS
   79 A(I,J) = 0.
      NC = NMBS + 1
C
C
C READ MEMBER INCIDENCES AND PROPERTIES
      DO 120 MB = 1,NMBS
      READ (5,80) M,JNTST,JNTEND,AREAM
   80 FORMAT (3I4, F12.4)
      IF (                         (JNTEND.LE.NJS).AND.(JNTST.LT.JNTEND)) GO
     * TO 90
      WRITE (6,85) M
   85 FORMAT (1H-,20H INCIDENCE OF MEMBER,I4,39H INCORRECTLY SPECIFIED.
     * JOB TERMINATED)
      CALL EXIT
   90 CONTINUE
      IF (AREAM.EQ.0.) AREAM = 1.
      AREA(M) = AREAM
C CALCULATE MEMBER LENGTHS
      TLGTH = 0.
      DO 95 I = 1,JF
   95 TLGTH = TLGTH + (COORD(JNTEND,I) - COORD(JNTST,I))**2
      LENGTH = SQRT(TLGTH)
      OBJ(M) = LENGTH
C
C CALCULATE MEMBER DIRECTION COSINES
      DO 100 I = 1,JF
  100 DRNCOS(I) = (COORD(JNTEND,I) - COORD(JNTST,I))/LENGTH
C
C OUTPUT MEMBER INCIDENCES AND PROPERTIES
      WRITE (6,105) M,JNTST,JNTEND,LENGTH,AREAM,(DRNCOS(I),I = 1,JF)
  105 FORMAT (1H , 23X, 3I11,2F12.4, 6E11.3)
C
C ADD MEMBER DIRECTION COSINES INTO THE CONNEXION MATRIX
```

```
      LROCS = JF*JNTST
      IRDCS = LRDCS - JF + 1
      DO 110 I = IROCS, LROCS
      J = I - IROCS + 1
  110 A(I,M) = -DRNCOS(J)
      LRDCE = JF*JNTEND
      IRDCE = LRDCE - JF +1
      DO 115 I = IROCE, LROCE
      J = I - IRDCE + 1
  115 A(I,M) = DRNCDS(J)
  120 CONTINUE
C
C
C READ SIZES OF LOAD COMPONENT AT EACH JOINT
C MUST NUMBER IN ORDER X, Y, Z, AT EACH JOINT
C NLJTS = NUMBER OF LOADED JOINTS
      READ (5,140) NLJTS
  140 FORMAT (16I4)
      DO 145 I = 1,NRC
  145 Q(I) = 0.
      DO 155 NO = 1,NLJTS
      READ (5,150) N, (QJ(I), I = 1,JF)
  150 FORMAT (I4,9F8.4)
      DO 155 I = 1,JF
      J = JF*(N-1) + I
  155 Q(J) = QJ(I)
C
C TO CORRECT MATRICES C AND Q FOR SUPPORTS
      NFJSP = NFJS + 1
      M = 0
      DO 157 I = NFJSP, NJS
      DO 157 J = 1, JF
      IF (RELS(I,J).NE.1.) GO TO 157
      K = JF*(I - 1) + J - M
      NRC = NRC - 1
      DO 156 L = K, NRC
      N = L + 1
      Q(L) = Q(N)
      DO 156 II = 1, NMBS
      A(L,II) = A(N,II)
  156 CONTINUE
      M = M + 1
  157 CONTINUE
C
C OUTPUT CONNEXION MATRIX
      WRITE (6,125)
  125 FORMAT ( 1H-, 18HCONNEXION MATRIX C/)
      DO 130 I = 1,NRC
  130 WRITE (6,135) (A(I,J),J = 1,NMBS)
  135 FORMAT (1H , 10E12.3)
  170 CONTINUE
      RETURN
      END


$IBFTC RANK 2
      SUBROUTINE RANK 2 (A,MRED,KNUT,MOP,NTMBS,NTCA,NTIV,NR,NCC,NRD)
      REAL A(NTMBS,NTCA)
```

```
      INTEGER MRED(NTIV),KNUT(NTIV),MOP(NTMBS)
C
C SET UP UNIT SUBMATRIX
      NCP = NCC + 1
      NC = NCC + NR
      DO 4 I = 1,NR
      DO 3 J = NCP,NC
    3 A(I,J) = 0.
      K = I + NCC
    4 A(I,K) = 1.
C
C JORDAN ELIMINATION
      DO 25 I = 1,NR
      DEN = 0.
      DO 5 J = 1,NCC
      AB = ABS(A(I,J))
      DN = ABS(DEN)
      IF (AB.GT.DN) JJ = J
      IF (AB.GT.DN) DEN = A(I,J)
    5 CONTINUE
      IF (DEN.EQ.0.) GO TO 25
      DO 10 K = 1,NC
   10 A(I,K) = A(I,K)/DEN
      DO 20 L = 1,NR
      IF (L.EQ.I) GO TO 20
      FAC = A(L,JJ)
      DO 15 M = 1,NC
   15 A(L,M) = A(L,M) - A(I,M)*FAC
   20 CONTINUE
   25 CONTINUE
  301 FORMAT (1H /(20F6.2))
C
C CHECK FOR INCONSISTENCY AND DEPENDENCE
      NDF = 0
      DO 65 I = 1,NR
      J = 0
   45 J = J + 1
      IF (J.GT.NC) GO TO 55
      IF (A(I,J).EQ.0.) GO TO 45
      IF (J.LE.NCC) GO TO 65
      NDF = NDF + 1
      GO TO 65
   55 NR = NR - 1
      DO 60 K = I,NR
      L = K + 1
      DO 60 M = I,NC
   60 A(K,M) = A(L,M)
   65 CONTINUE
      IF (NDF.EQ.0) GO TO 66
      WRITE (6,50) NDF
   50 FORMAT (1H-,28HTHE STRUCTURE IS A MECHANISM/30HNUMBER OF DEGREES OF
     *F FREEDOM =,I3)
   66 CONTINUE
C
C ISOLATE REDUNDANCIES
      NRD = 0
      DO 75 J = 1,NCC
      KOUNT = 0
      DO 70 I = 1,NR
   70 IF (A(I,J).NE.0.) KOUNT = KOUNT + 1
```

```
      IF (KOUNT.LE.1) GO TO 75
      NRD = NRD + 1
      MRED(NRD) = J
   75 CONTINUE
C
C REDUNDANCIES ... CHECK, OUTPUT
      I = NCC - NR
      IF (I.NE.NRD) WRITE (6,80)
   80 FORMAT (1H-,22H(COLS - ROWS) .NE. NRD)
      IF (NRD) 85,85,95
   85 WRITE (6,90)
   90 FORMAT (1H0,32HSTRUCTURE IS ALREADY DETERMINATE)
      GO TO 105
   95 WRITE (6,100) NRD,(MRED(I),I = 1,NRD)
  100 FORMAT (1H0,13HSTRUCTURE IS ,I2,16H TIMES REDUNDANT/
     * 22H REDUNDANT MEMBERS ARE,40I3)
  105 CONTINUE
C
C MOVE ALL REDUNDANT COLUMNS TO RHS OF MATRIX A
      IF (NRD.LE.0) GO TO 135
      DO 110 K = 1,NRD
  110 KNUT(K) = 0
      DO 130 I = 1,NRD
      IF (MRED(I).GT.NR) GO TO 130
      J = NR
  115 J = J + 1
      K = 0
  120 K = K + 1
      IF (MRED(K).EQ.J) GO TO 115
      IF (K.LT.NRD) GO TO 120
      K = MRED(I)
      DO 125 L = 1,NR
      BLOC = A(L,J)
      A(L,J) = A(L,K)
      A(L,K) = BLOC
  125 CONTINUE
      KNUT(I) = MRED(I)
      MRED(I)= J
  130 CONTINUE
  135 CONTINUE
C
C FORM UNIT MATRIX IN A(NR,NR)
      DO 155 LOO = 1,2
      DO 140 I = 1,NR
      DO 140 J = 1,NR
      IF (A(I,J).EQ.0.) GO TO 140
      MOP(I) = J
  140 CONTINUE
      DO 155 I = 1,NR
      IF (MOP(I).EQ.I) GO TO 155
      DO 145 K = I,NR
      IF (MOP(K).EQ.I) L=K
  145 CONTINUE
      DO 150 J = 1,NC
      BLOC = A(I,J)
      A(I,J) = A(L,J)
      A(L,J) = BLOC
  150 CONTINUE
      MOP(L) = MOP(I)
  155 CONTINUE
```

```
C
C INCREASE A TO GET COMPLETE INTERNAL LOAD SYSTEM
      K = NR + 1
      IF (NRD.LE.0) GO TO 170
      DO 165 I = K,NCC
      DO 160 J = K,NC
  160 A(I,J) = 0.
  165 A(I,I) = -1.
  170 CONTINUE
C
C BACK TO ORIGINAL ORDER OF ELEMENT FORCES
      IF (NRO.EQ.0) GO TO 174
      DO 173 I = 1,NRD
      IF (KNUT(I).EQ.0) GO TO 173
      J = KNUT(I)
      M = MRED(I)
      MRED(I) = J
      DO 172 L = K,NC
      BLOC = A(J,L)
      A(J,L) = A(M,L)
      A(M,L) = BLOC
  172 CONTINUE
  173 CONTINUE
  174 CONTINUE
      RETURN
      END


$IBFTC HAAR
      SUBROUTINE HAAR (Q,AREA,A,QL,SENS,X,MRED,NTMBS,NTCA,NTRD,NMBS,NRD,
     * NRC,ALF,ISW)
      REAL Q(NTMBS),A(NTMBS,NTCA),QL(NTMBS),SENS(NTMBS,NTRD),
     *              AREA(NTMBS),X(NTRD)
      INTEGER MRED(NTRD)
C
C FORM SENSITIVITIES
      DO 10 I = 1,NRD
      K = NMBS - NRD + I
      L = MRED(I)
      DO 10 J = 1,NMBS
   10 SENS(J,I) = A(J,K)*AREA(L)/AREA(J)
C
C INCLUDE LOAD FACTOR FAC - READ IT IF SSWTCH 5 ON
      IF (ISW.EQ.2) GO TO 12
      READ (5,11) FAC
   11 FORMAT (10F8.4)
      GO TO 13
   12 FAC = 1.
   13 DO 14 I = 1,NRC
   14 Q(I) = Q(I)*FAC
C
C FORM QL MATRIX
      DO 15 I = 1,NMBS
      QL(I) = 0.
      DO 15 J = 1,NRC
      K = J + NMBS
   15 QL(I) = QL(I) + A(I,K)*Q(J)/AREA(I)
C
```

```
C FORM X(INITIAL), LHS, RHS
      DO 25 I = 1,NRD
   25 X(I) = 1.
C
      WRITE (6,610) (Q(I),I = 1,NRC)
  610 FORMAT (1H0,10HC MATRIX -/(10E12.3))
      WRITE (6,611) (CL(I),I = 1,NMBS)
  611 FORMAT (1H0,11HCL MATRIX -/(10E12.3))
      WRITE(6,602)((SENS(I,J),I=1,NMBS),J=1,NRD)
  602 FORMAT(1H0,6HSENS -/(10E12.3))
      RETURN
      END



$IBFTC KARMAN
      SUBROUTINE KARMAN (LEN,AREA,QL,SENS,OBJ,X,LHS,RHS,NTMBS,NTRD,NRD,
     * NMBS,ALF)
      REAL LEN(NTMBS),AREA(NTMBS),CL(NTMBS),SENS(NTMBS,NTRD),OBJ(NTRD),
     * X(NTRD),LHS(NTMBS),RHS(NTMBS)
C
C FORM OBJECTIVE COEFFICIENTS
      DO 15 I = 1,NRD
      OBJ(I) = 0.
      DO 15 J = 1,NMBS
      BLOC = 0.
      DO 10 K = 1,NRD
   10 BLOC = BLOC + (X(K) - 1.)*SENS(J,K)
   15 OBJ(I) = OBJ(I) - 2.*(QL(J) + BLOC)*LEN(J)*SENS(J,I)*AREA(J)
      IF ((NRD.EQ.1).AND.(OBJ(1).GE.0.)) OBJ(1) = +1.
      IF ((NRD.EQ.1).AND.(OBJ(1).LT.0.)) OBJ(1) = -1.
C
C COMPUTE CONSTRAINTS
      DO 25 I = 1,NMBS
      BLOC = 0.
      DO 20 J = 1,NRD
   20 BLOC = BLOC + SENS(I,J)*(1.- X(J))
      LHS(I) = BLOC - ALF - QL(I)
   25 RHS(I) = BLOC + 1. - QL(I)
      RETURN
      END



$IBFTC DUALP5
      SUBROUTINE DUALP5 (SENS,OBJ,LHS,RHS,X,Z,IPRNT,CHS,KUT,IDEG,NAME,MARK
     * RK,BOX,ADAP,NTRS,NTMBS,NRZ,NCZ,M,N,OPT1,OPT2,INDEX)
C
C          LINEAR PROGRAMMING BY SOLUTION OF DUAL
C
C          OUTPUT INDEX 'IPRNT' FOR THIS SUBROUTINE
C           IPRNT = 0  NO OUTPUT
C           IPRNT = 1  OUTPUT IFAS,MOVE LIMITS
C           IPRNT = 2  OUTPUT IFAS,MOVE LIMITS,Z-ARRAYS ETC.
C           IPRNT = 3  AS FOR 2 PLUS D-ROW OUTPUT FROM KRSIMP
C
      REAL SENS(NTRS,NTMBS),OBJ(NTMBS),LHS(NTRS),RHS(NTRS),Z(NRZ,NCZ),
     * CHS(NTMBS),X(NTMBS),BOX(NTMBS),OPT2
```

```
      INTEGER KUT(NCZ),IDEG(NCZ),NAME(NCZ),MARK(NRZ),ADAP(NTMBS,2),OPT1
C
C          CLEAR Z-ARRAY
      DO 100 I = 1,NRZ
      DO 100 J = 1,NCZ
  100 Z(I,J) = 0.
C          SET DUAL SENSITIVITIES, OBJECTIVE COEFFICIENTS
C          AND ACTIVITY CODES IN Z-ARRAY
      ND = 0
      DO 113 J = 1,M
  110 ND = ND + 1
       DO 111 I = 1,N
      Z(I,J) = SENS(J,I)
      KK = M + ND
  111 Z(I,KK) = -SENS(J,I)
      Z(N+1,J) = -RHS(J)
      Z(N+1,KK) = LHS(J)
  113 CONTINUE
C          SET COLUMNS FOR MOVE LIMITS IN Z-ARRAY
      GO TO (120,114,114),OPT1
  114 IF (INDEX.NE.1) GO TO 116
      DO 115 I = 1,N
      ADAP(I,1) = 0
  115 BOX(I) = 1.
  116 DO 119 I = 1,N
      T = .01*OPT2*X(I)*BOX(I)
      NM = M + ND + I
      Z(I,NM) = 1.
      Z(N+1,NM) = -T
      NM = NM + N
      Z(I,NM) = -1.
  119 Z(N+1,NM) = -T
      IF (OPT1.NE.3.OR.IPRNT.LT.1) GO TO 120
      WRITE (6,118) (BOX(I),I=1,N)
  118 FORMAT(20HOCURRENT MOVE LIMITS,10F10.3)
  120 CONTINUE
C          SET DUAL L.H. AND R.H. SIDES AND
C          CONSTRAINT CODES IN Z-ARRAY
      GO TO (121,122,122),OPT1
  121 NZ = M + ND
      GO TO 123
  122 NZ = M + ND + N + N
  123 DO 130 I = 1,N
      Z(I,NZ+3) = OBJ(I)
  125 Z(I,NZ+1) = 3.
  130 CONTINUE
C          MAKE R.H.SIDES OF DUAL CONSTRAINTS POSITIVE
      DO 135 I = 1,N
      CHS(I) = -1.
      IF (Z(I,NZ+3)) 132,135,135
  132 CHS(I) = 1.
      DO 133 J = 1,NZ
  133 Z(I,J) = -Z(I,J)
      Z(I,NZ+3) = -Z(I,NZ+3)
      IF (Z(I,NZ+1).EQ.3.) GO TO 135
      Z(I,NZ+1) = 1.
  135 CONTINUE
C          OUTPUT Z-ARRAY BEFORE CALLING KRANTE
      MZ = N
      MZ2 = N + 2
```

```
      NZ3 = NZ + 3
      IF (IPRNT.LT.2) GO TO 140
      WRITE (6,136)
  136 FORMAT(30H0Z-ARRAY BEFORE CALLING KRANTE/)
      DO 137 I = 1,MZ2
  137 WRITE (6,138) (Z(I,J),J=1,NZ3)
  138 FORMAT(1H ,10E12.3)
  140 CONTINUE
      CALL KRANTE (Z,MZ,NZ,IFAS,KUT,NRZ,NCZ)
C           OUTPUT Z-ARRAY ETC. BEFORE CALLING KRSIMP
      MZ2 = MZ + 2
      NZ1 = NZ + 1
      IF (IPRNT.LT.2) GO TO 145
      WRITE (6,141)
  141 FORMAT(30H0Z-ARRAY BEFORE CALLING KRSIMP/)
      DO 142 I = 1,MZ2
  142 WRITE (6,138) (Z(I,J),J=1,NZ1)
      WRITE (6,144) IFAS,(KUT(J),J=1,NZ)
  144 FORMAT(7H0IFAS =,I4/12H0KUT ARRAY -/(1H ,24I5))
  145 CONTINUE
      CALL KRSIMP (Z,IFAS,MZ,NZ,NAME,IDEG,KUT,NRZ,NCZ,IPRNT,MARK)
      IF (IFAS.EQ.3) GO TO 162
C           OUTPUT Z-ARRAY ON RETURN FROM KRSIMP
      MZ1 = MZ + 1
      IF (IPRNT.LT.2) GO TO 150
      WRITE (6,146)
  146 FORMAT (30H0Z-ARRAY ON RETURN FROM KRSIMP/)
      DO 147 I = 1,MZ1
  147 WRITE (6,138) (Z(I,J),J=1,NZ1)
  150 CONTINUE
      IF (IPRNT.LT.1) GO TO 152
      WRITE (6,151) IFAS
  151 FORMAT (7H0IFAS =,I4)
  152 CONTINUE
C           SET VALUES OF PRIMAL VARIABLES IN FIRST ROW
C           OF Z-ARRAY, PRIMAL OBJECTIVE IN Z(2,1)
      DO 154 J = 1,MZ
  154 Z(1,J) = CHS(J) * Z(MZ1,J)
      Z(2,1) = -Z(2,1)
C           CHECK FOR ADAPTION OF MOVE LIMITS
      GO TO (160,160,156),OPT1
  156 DO 158 I = 1,N
      ADAP(I,2) = ADAP(I,1)
      ADAP(I,1) = -1
      IF (Z(1,I).GT.0.) ADAP(I,1) = 1
  158 IF (IABS(ADAP(I,1) - ADAP(I,2)).GT.1) BOX(I) = 0.5 * BOX(I)
  160 CONTINUE
      RETURN
  162 Z(2,1) = 0.0
      RETURN
      END


$IBFTC DEFLN
      SUBROUTINE DEFLN (A,OBJ,PROD,NYLD,D,DLGTH,C,MRED,KNUT,MOP,NTMBS,
     * NTIV,NTCA,NMBS,NRD,NRC,ISW,ALF)
      REAL A(NTMBS,NTCA),OBJ(NTMBS),PROD(NTMBS),D(NTMBS),DLGTH(NTMBS),
     * C(NTMBS,NTMBS)
```

```
      INTEGER NYLD(NTMBS),MRED(NTIV),KNUT(NTIV),MOP(NTMBS)
C
C SET UP SET OF YIELDED MEMBERS
      GO TO (4,6,3),ISW
    3 NRD = 0
      NY = 0
      GO TO 11
    4 READ (5,5) NRD, NY, (NYLD(I),I = 1,NY)
    5 FORMAT (16I4)
      GO TO 11
    6 J = 0
      DO 10 I = 1,NMBS
      APRO = PROD(I)
      IF (APRO.LT.0.) GO TO 8
      IF ((APRO.LT..9999).OR.(APRO.GT.1.0001)) GO TO 10
      GO TO 9
    8 IF ((APRO.LT.(-ALF-.0001)).OR.(APRO.GT.(-ALF+.0001))) GO TO 10
    9 J = J + 1
      NYLD(J) = I
   10 CONTINUE
      NY = J
      IF (NY.LT.NRD) NRD = NY
   11 CONTINUE
C
C CALCULATE THE YIELDED MEMBERS TO BE ELIMINATED
      NIV = NRD + 1
      NEL = NY - NRD
      WRITE (6,15) NEL
   15 FORMAT (1H-,5HNEL =,I3)
      IF (NEL.GT.4) RETURN
C
      IG = NRD + 4
      IE = NRD + 3
      ID = NRD + 2
      IC = NRD + 1
C
      DO 70 I4 = 1,IG
      IF (I4.GE.ID) GO TO 60
C
      IH = NRD + 3 - I4
      DO 55 I3 = 1,IE
      IF (I3.GE.IH) GO TO 45
C
      IB = NRD + 4 - I3 - I4
      DO 40 I2 = 1,IC
      IF (I2.GE.IB) GO TO 30
C
      IA = NRD + 4 - I2 - I3 - I4
      DO 25 I1 = 1,IA
C
C
C
C PUTTING NYLD(1 TO NRD) INTO ASCENDING ORDER
      MA = I2 + I3 + I4 - 3
      MB = MA + 1
      IF (MA.EQ.0) GO TO 215
      DO 210 II = 1,MA
      BLOK = NYLD(1)
      DO 205 JJ = 2,NRD
      KK = JJ - 1
```

```
  205 NYLD(KK) = NYLC(JJ)
      NYLD(NRD) = BLCK
  210 CONTINUE
  215 CONTINUE
C
      JF = I1 - 1
      IF (MB.GE.NRD) GO TO 218
      IF (JF.EQ.0) GC TO 218
      DO 217 JG = 1,JF
      BLOC = NYLD(1)
      JH = NRD - MA
      DO 216 JJ = 2,JH
      JK = JJ - 1
  216 NYLD(JK) = NYLC(JJ)
      NYLD(JH) = BLOC
  217 CONTINUE
  218 CONTINUE
C
C FORM DETERMINATE C MATRIX
      DO 3CO I = 1,NRC
      DO 300 J = 1,NMBS
  300 A(I,J) = C(I,J)
C
C ELIMINATING YIELDEC MEMBERS
  219 FORMAT (1H /(2CF6.2))
      NM = NMBS
      NMM = NMBS - 1
      IF (NRD.EQ.0) GO TO 325
      DO 324 II = 1,NRD
      JJ = NYLD(II) - II + 1
      NM = NM - 1
      DO 322 MM = 1,NRC
      DO 322 KK = JJ,NM
      LL = KK + 1
  322 A(MM,KK) = A(MM,LL)
      BLOA = PROD(JJ)
      BLOB = OBJ(JJ)
      DO 323 KK = JJ,NMM
      LL = KK + 1
      PROD(KK) = PROC(LL)
  323 OBJ(KK) = OBJ(LL)
      PROD(LL) = BLOA
      OBJ(LL) = BLOB
  324 CONTINUE
      WRITE (6,330) (NYLD(IQ),IQ = 1,NRD)
  330 FORMAT (1HO,22HELIMINATED MEMBERS ARE,16I4)
      IF (NY.LE.NRD) GO TO 340
      IQQ = NRD + 1
      WRITE (6,335) (NYLD(IQ),IQ = IQQ,NY)
  335 FORMAT (1HO,25HLAST TO YIELD ARE MEMBERS,16I4)
  340 CONTINUE
  325 CONTINUE
C
C FORM BO MATRIX
      NRCE = NRC
      NEM = NM
      CALL RANK 2 (A,MRED,KNUT,MOP,NTMBS,NTCA,NTIV,NRCE,NEM,NOO)
C
C CALCULATE DEFLECTIONS
C OUTPUT DEFLEXIONS
```

```
      KK = II + NM
      D(II) = 0.
      DO 225 JJ = 1,NM
  225 D(II) = D(II) + A(JJ,KK)*OBJ(JJ)*PROD(JJ)/10.
  245 FORMAT (1H0,15HVECTOR NYLD IS ,16I4)
      WRITE (6,250) (D(IQ),IQ = 1,NRC)
  250 FORMAT (1H0,18HJOINT DEFORMATIONS/(10E12.3))
C
C CALCULATE DISTORTED MEMBER LENGTHS
      DO 275 N1 = 1,NMBS
      DLGTH(N1) = 0.
      DO 275 N2 = 1,NRC
  275 DLGTH(N1) = DLGTH(N1) + C(N2,N1)*D(N2)
      WRITE (6,270) (DLGTH(N3),N3 = 1,NMBS)
  270 FORMAT (1H ,22HDISTORTED LENGTHS ARE /(10F7.3))
C
C CHECK ON JOINT DEFLEXIONS
      IF (NOO.EQ.0) GO TO 285
      DO 279 II = 1,NOO
      KK = NM + 1 - II
      D(II) = 0.
      DO 279 JJ = 1,NM
  279 D(II) = D(II) + A(JJ,KK)*OBJ(JJ)*PROD(JJ)/10.
      WRITE (6,280) (D(II),II = 1,NOO)
  280 FORMAT (1H ,32HRELATIVE REDUNDANT DISPLACEMENTS/(10E11.3))
  285 CONTINUE
C
C RETURNING MATRICES SENS, PROD, OBJ, TO INITIAL ORDER
      IF (NRD.EQ.0) GO TO 350
      DO 349 II = 1,NRD
      JJ = NRD + 1 - II
      KK = NYLD(JJ) - JJ + 2
      NM = NM + 1
      BLOA = PROD(NMBS)
      BLOB = OBJ(NMBS)
      DO 348 LL = KK,NMBS
      IIA = NMBS + KK - LL
      IIB = IIA - 1
      PROD(IIA) = PROD(IIB)
  348 OBJ(IIA) = OBJ(IIB)
      PROD(IIB) = BLOA
      OBJ(IIB) = BLOB
  349 CONTINUE
  350 CONTINUE
C
C COMPUTE THE RATIOS OF ACTUAL STRAIN TO YIELD SYRAIN
      DO 290 N1 = 1,NMBS
      DLGTH(N1) = SIGN(DLGTH(N1),PROD(N1))
  290 DLGTH(N1) = DLGTH(N1)*10./OBJ(N1)
      WRITE (6,295) (DLGTH(N3),N3 = 1,NMBS)
  295 FORMAT (1H ,52HRATIO OF ACTUAL MEMBER STRAIN TO MEMBER YIELD STRAI
     *N/(10F7.3))
C
C PUT NYLD(1 TO NRD) INTO FORMER ORDER
      IF (MB.GE.NRD) GO TO 228
      IF (JF.EQ.0) GO TO 228
      DO 227 JG = 1,JF
      BLOC = NYLD(JH)
      JI = JH - 1
```

```
      DO 226 JJ = 1,JI
      JK = JH + 1 - JJ
      JL = JK - 1
  226 NYLD(JK) = NYLD(JL)
      NYLD(1) = BLOC
  227 CONTINUE
  228 CONTINUE
C
      IF (MA.EQ.0) GO TO 240
      DO 235 II = 1,MA
      BLOK = NYLD(NRD)
      IN = NRD - 1
      DO 230 JJ = 1,IN
      KK = NRD + 1 - JJ
      LL = KK - 1
  230 NYLD(KK) = NYLD(LL)
      NYLD(1) = BLOK
  235 CONTINUE
  240 CONTINUE
C
C
C
      IF (NEL.LE.0) GO TO 200
      BLOC = NYLD(IC)
      JA = I2 + I3 + I4 - 2
      JB = IA - 1
      DO 20 JC = 1,JB
      JD = NRD + 2 - JC
      JE = JD - 1
   20 NYLD(JD) = NYLD(JE)
      NYLD(JA) = BLOC
   25 CONTINUE
      IF (NEL.EQ.1) GO TO 200
C
   30 BLOC = NYLD(ID)
      DO 35 KA = 1,IC
      KB = NRD + 3 - KA
      KC = KB - 1
   35 NYLD(KB) = NYLD(KC)
      NYLD(1) = BLOC
   40 CONTINUE
      IF (NEL.EQ.2) GO TO 200
C
   45 BLOC = NYLD(IE)
      DO 50 LA = 1,ID
      LB = NRD + 4 - LA
      LC = LB - 1
   50 NYLD(LB) = NYLD(LC)
      NYLD(1) = BLOC
   55 CONTINUE
      IF (NEL.EQ.3) GO TO 200
C
   60 BLOC = NYLD(IG)
      DO 65 NA = 1,IE
      NB = NRD + 5 - NA
      NC = NB - 1
   65 NYLD(NB) = NYLD(NC)
      NYLD(1) = BLOC
   70 CONTINUE
  200 CONTINUE
```

```
C
      RETURN
      END


$IBFTC NOWEND
      SUBROUTINE NOWEND (Z,QL,SENS,PROD,LEN,AREA,X,NTMBS,NRZ,NCZ,NTRD,
     * NMBS,NRD,KOUNT)
      REAL Z(NRZ,NCZ),QL(NTMBS),SENS(NTMBS,NTRD),PROD(NTMBS),LEN(NTMBS),
     * AREA(NTMBS),X(NTRD)
C
C OUT PUT OPTIMUMM VALUES OF LAMDA
      WRITE (6,10)
   10 FORMAT (1H-,50H**************************************************)
      WRITE (6,11) KCUNT
   11 FORMAT (1H0,16HITERATION NUMBER,I4)
      WRITE (6,15) Z(2,I)
   15 FORMAT (1H-,15HOPTIMUM LAMDA -,1PE12.4)
C
COUTPUT MEMBER FORCES
      DO25 I = 1,NMBS
      PROD(I) = 0.
      DO 20 J = 1,NRD
   20 PROD(I) = PROD(I) + SENS(I,J)*(X(J) - 1.)
   25 PROD(I) = PROD(I) + QL(I)
      WRITE (6,30) (PROD(I),I = 1,NMBS)
   30 FORMAT (1H0, 16HPRODUCT MATRIX -/(6E12.4))
      WRITE (6,50) (Z(1,I),I = 1,NRD)
   50 FORMAT (1H0, 7HZ ROW -/(10E12.4))
      WRITE (6,55) (X(I),I = 1,NRD)
   55 FORMAT (1H0,17HX VARIABLES ARE -/(10E12.4))
C
C CALCULATE STRAIN ENERGY
      SE = 0.
      DO 40 I = 1,NMBS
   40 SE = SE + LEN(I)*AREA(I)*(PROD(I)**2)
      WRITE (6,45) SE
   45 FORMAT (1H0,16HSTRAIN ENERGY IS,E12.4)
C
      RETURN
      END
```

APPENDIX C

Appendix C presents the listings of six of the eight subroutines of the program MULTI-LOAD PLASTIC DESIGN. This is a program to determine the minimum weight designs of two- or three-dimensional pin-jointed trusses, with ideal elastic-plastic member load-deformation behaviours, for one or several loading cases.

The program is dimensioned:

maximum number of joints $j = 16$
maximum number of members $m = 30$
maximum number of redundants $r = 10$
maximum number of loading cases $c = 10$

(Note, that in equations 5.7 and 5.8, the maximum number of variables = *40* and the maximum number of constraints = *60*).

The relevant matrices of the program are:

COORD	is the	matrix of joint coordinates
RELS	is the	matrix of support restraints
A	is the	augmented matrix $[C \vdots I]$
Z	is the	Simplex tableau
ACT	is the	vector of the signs of the variables
SENS	is the	coefficient matrix (equation 5.8)
RHS	is the	vector of right hand side constraints (equation 5.8)
OBJ	is the	member length vector, augmented $[\ell^T \vdots 0^T]$
Q	is the	joint load vector p
MRED	is the	vector of redundant member numbers
ENDS	is the	matrix of the member incidence joint numbers
ALF	is the	compression coefficient α

The two subroutines KRANTE and KRSIMP describe a two-phase Standard Simplex algorithm and are not listed.

```
$JOB            357361, MARKS
$TIME           6C
$*              MULTI-LOAD PLASTIC DESIGN   (TOAKLEY)
$IBJOB          MAP
$IBFTC POLOAD   NODECK
C PROGRAM FOR DETERMINING OPTIMUM (MINIMUM VOLUME) CROSS-SECTIONAL AREAS
C  OF THREE-DIMENSIONAL TRUSSES, FOR GIVEN LOAD AND CONFIGURATION
C THE PROGRAM DESIGNS FOR SEVERAL LOAD CASES.
C THE PROGRAM IS COMPRISED OF EIGHT SUB-PROGRAMS -
C       POLOAD - THE VARIABLE ARRAYS AND CALL STATEMENTS
C       PLOPT5 - FORMS THE MATHEMATICAL MODAL
C       RANK 2 - ISOLATES THE REDUNDANTS
C       MANYLD - SETS UP THE L.P. VALUES
C       DUALP  - FORMS THE Z ARRAY
C       KRANTE + KRSIMP - THE L. P. SUB-PROGRAMS
C.      FINAL  - OUTPUTS THE RESULTS
C
      REAL COORD(16,3),RELS(16,3),A(30,60),Z( 70,160),ACT(40),
     * SENS( 60,40),RHS( 60),OBJ(40),CHS(40),Q(30,10)
      INTEGER KUT(160),IDEG(160),NAME(160),MARK( 70),MRED(10),MOP(30),
     * KNUT(10),ENDS(30,2)
C
      DO 170 LOOK = 1,20
      CALL TIME
      CALL PLOPT5 (COORD,RELS,A,Q,OBJ,ENDS,16,30,10,60,40,NRC,NMBS,NLDS,
     * JF,NJS,NSJS)
      CALL RANK 2 (A,MRED,KNUT,MOP,30,60,10,NRC,NMBS,NRD)
      CALL MANYLD (A,Q,OBJ,SENS,ACT,RHS,60,40,30,60,10,NMBS,NRD,NIV,NRS,
     * NLDS,NRC)
      CALL DUALP2 (SENS,OBJ,ACT,RHS,Z,1,CHS,KUT,IDEG,NAME,MARK, 60,40,
     *  70,160,NRS,NIV)
      CALL FINAL (Z, 70,160,NMBS,NIV)
      CALL TIME (NM,NS,NSS)
C
      WRITE (6,169) NM, NS, NSS
  169 FORMAT(1H-,32HTIME TAKEN FOR ABOVE STRUCTURE -/20X,I4,5H MINS,I6,
     * 5H SECS,I6,6H SSECS)
  170 CONTINUE
      CALL EXIT
      END



$IBFTC PLOPT5
      SUBROUTINE PLOPT5 (COORD,RELS,A,Q,OBJ,ENDS,NTJS,NTMBS,NTLDS,NTCA,
     * NTIV,NRC,NMBS,NLDS,JF,NJS,NSJS)
      REAL COORD(NTJS,3),RELS(NTJS,3),A(NTMBS,NTCA),TYPE(2),XYZ(3),
     * PRNAME(13),ORNCOS(3),QJ(3),LENGTH,Q(NTMBS,NTLDS)
      REAL OBJ(NTIV)
      INTEGER ENDS(NTMBS,2)
      DATA TYPE/6H PLANE, 6H SPACE/, XYZ/1HX, 1HY, 1HZ/, FIN/6HFINISH/
C
C READ HEADER CARD
      READ (5,10) PRNAME
   10 FORMAT (13A6)
      IF (PRNAME(1).EQ.FIN) CALL EXIT
      WRITE (6,15) PRNAME
   15 FORMAT (1H1, 13A6)
C
```

```
C READ PROBLEM PARAMETERS -
C    JF = 2 FOR TWO-DIMENSIONAL TRUSS
C    JF = 3 FOR THREE-DIMENSIONAL TRUSS
C   NJS = NUMBER OF JOINTS IN TRUSS
C  NSJS = NUMBER OF SUPPORT JOINTS IN TRUSS
C  NMBS = NUMBER OF MEMBERS IN TRUSS
      WRITE (6,20)
   20 FORMAT (1H0, 10X, 6HDATA -)
      READ(5,25) JF, NJS, NSJS, NMBS
   25 FORMAT (20I4)
      NFJS = NJS - NSJS
C
C OUTPUT PROBLEM PARAMETERS
      WRITE (6,30) TYPE(JF - 1)
   30 FORMAT (1H0,20X,20HTYPE OF STRUCTURE  -,A6,29H TRUSS WITH THE CROSS-S
     *S-SECTION/42X,27HAREA OF EACH MEMBER VARYING)
      WRITE (6,35) NJS, NSJS, NMBS
   35 FORMAT (1H0,20X,20HNUMBER OF JOINTS   =,I4/31X,10HSUPPORTS =,I4/3
     *1X,10HMEMBERS  =,I4)
C
C READ JOINT NUMBERS AND COORDINATES -
C FREE JOINTS MUST BE NUMBERED FIRST, THEN SUPPORTS
C IF RELS(N,1) = 1. ... RESTRAINT AT N IN X-DIRN.
C IF RELS(N,2) = 1. ... RESTAINT AT N IN Y-DIRN.
C IF RELS(N,3) = 1. ... RESTRAINT AT N IN Z-DIRN.
      DO 39 I = 1,NJS
      DO 39 J = 1,JF
   39 RELS(I,J) = 0.
      DO 40 I = 1,NJS
      READ (5,45) N, (COORD(N,J),J = 1,JF)
   45 FORMAT (I4, 3F8.4)
      IF (N.LE.NFJS) GO TO 40
      READ (5,46) (RELS(N,J),J = 1,JF)
   46 FORMAT (3F4.1)
   40 CONTINUE
C
C OUTPUT JOINT NUMBERS AND COORDINATES - FIRST FOR FREE JOINT AND THEN
C SUPPORT JOINTS
      WRITE (6,50) (XYZ(I),I=1,JF)
   50 FORMAT (1H0,20X,20HJOINT COORDINATES  -//28X,5HJOINT,10X,3(A1,11X))
     *)
      WRITE (6,55)
   55 FORMAT (23X,4HFREE)
      DO 60 I = 1,NJS
      WRITE (6,65) I, (COORD(I,J),J=1,JF)
   60 IF (I.EQ.NFJS) WRITE (6,70)
   65 FORMAT(1H ,I31,2X,6F12.4)
   70 FORMAT (23X,7HSUPPORT)
      NFJSP = NFJS + 1
      WRITE (6,47) ((RELS(I,J),J = 1,JF),I = NFJSP,NJS)
   47 FORMAT (1H0,20X,6HRELS -/27X,(6F5.1))
      WRITE (6,75) (XYZ(I),I = 1,JF)
   75 FORMAT (1H0,20X,17HMEMBER DETAILS  -//31X,6HMEMBER,6X,5HSTART,7X,
     * 3HEND,5X,6HLENGTH,3(3X,7HDRNCOS-,A1))
C
C LOOP ENTERED FOR ALL MEMBERS
      NRC = JF*NJS
      DO 79 I = 1,NRC
      DO 79 J = 1,NMBS
   79 A(I,J) = 0.
```

```
      NC = NMBS + 1
C
C
C READ MEMBER INCIDENCES AND PROPERTIES
      DO 120 MB = I,NMBS
      READ (5,80) M,JNTST,JNTEND
   80 FORMAT (3I4, F12.4)
      IF (                        (JNTEND.LE.NJS).AND.(JNTST.LT.JNTEND)) GO
     * TO 90
      WRITE (6,85) M
   85 FORMAT (1H-,20H INCIDENCE OF MEMBER,I4,39H INCORRECTLY SPECIFIED.
     * JOB TERMINATED)
      CALL EXIT
   90 CONTINUE
      ENDS(M,1) = JNTST
      ENDS(M,2) = JNTEND
C CALCULATE MEMBER LENGTHS
      TLGTH = 0.
      DO 95 I = 1,JF
   95 TLGTH = TLGTH + (COORD(JNTEND,I) - COORD(JNTST,I))**2
      LENGTH = SQRT(TLGTH)
      DBJ(M) = LENGTH
C
C CALCULATE MEMBER DIRECTION COSINES
      DO 100 I = 1,JF
  100 DRNCOS(I) = (CCORD(JNTEND,I) - COORD(JNTST,I))/LENGTH
C
C OUTPUT MEMBER INCIDENCES AND PROPERTIES
      WRITE (6,105) M,JNTST,JNTEND,LENGTH,(DRNCOS(I),I = 1,JF)
  105 FORMAT (1H ,23X,3I11,F12.4,3E11.3)
C
C ADD MEMBER DIRECTION COSINES INTO THE CONNEXION MATRIX
      LROCS = JF*JNTST
      IROCS = LROCS - JF + 1
      DO 110 I = IROCS, LROCS
      J = I - IROCS + 1
  110 A(I,M) = -DRNCOS(J)
      LROCE = JF*JNTEND
      IROCE = LROCE - JF +1
      DO 115 I = IROCE, LROCE
      J = I - IROCE + 1
  115 A(I,M) = DRNCOS(J)
  120 CONTINUE
C
C
C READ SIZES OF LOAD COMPONENT AT EACH JOINT
C MUST NUMBER IN ORDER X, Y, Z, AT EACH JOINT
C NLJTS = NUMBER OF LOADED JOINTS
C NLDS = NUMBER OF LOADING CASES
      READ (5,140) NLDS
      DO 154 LOAD = 1,NLDS
      READ (5,140) NLJTS
  140 FORMAT (16I4)
      DO 145 I = 1,NRC
  145 Q(I,LOAD) = C.
      DO 155 NO = 1,NLJTS
      READ (5,150) N, (QJ(I), I = 1,JF)
  150 FORMAT (I4,9F8.4)
      DO 155 I = 1,JF
      J = JF*(N-1) + I
```

```
  155 C(J,LOAD) = QJ(I)
  154 CONTINUE
C
C TO CORRECT MATRICES C AND G FOR SUPPORTS
      NFJSP = NFJS + 1
      M = 0
      DO 157 I = NFJSP, NJS
      DO 157 J = 1, JF
      IF (RELS(I,J).NE.1.) GO TO 157
      K = JF*(I - 1) + J - M
      NRC = NRC - 1
      DO 156 L = K, NRC
      N = L + 1
      DO 156 II = 1, NMBS
      A(L,II) = A(N,II)
  156 CONTINUE
      M = M + 1
  157 CONTINUE
C
C OUTPUT CONNEXION MATRIX
      WRITE (6,125)
  125 FORMAT ( 1H-, 18HCONNEXION MATRIX C/)
      DO 130 I = 1,NRC
  130 WRITE (6,135) (A(I,J),J = 1,NMBS)
  135 FORMAT (1H , 10E12.3)
  170 CONTINUE
      RETURN
      END


$IBFTC RANK 2
      SUBROUTINE RANK 2 (A,MRED,KNUT,MOP,NTMBS,NTCA,NTIV,NR,NCC,NRD)
      REAL A(NTMBS,NTCA)
      INTEGER MRED(NTIV),KNUT(NTIV),MOP(NTMBS)
C
C SET UP UNIT SUBMATRIX
      NCP = NCC + 1
      NC = NCC + NR
      DO 4 I = 1,NR
      DO 3 J = NCP,NC
    3 A(I,J) = 0.
      K = I + NCC
    4 A(I,K) = 1.
C
C JORDAN ELIMINATION
      DO 25 I = 1,NR
      DEN = 0.
      DO 5 J = 1,NCC
      AB = ABS(A(I,J))
      DN = ABS(DEN)
      IF (AB.GT.DN) JJ = J
      IF (AB.GT.DN) DEN = A(I,J)
    5 CONTINUE
      IF (DEN.EQ.0.) GO TO 25
      DO 10 K = 1,NC
   10 A(I,K) = A(I,K)/DEN
      DO 20 L = 1,NR
      IF (L.EQ.I) GO TO 20
```

```
      FAC = A(L,JJ)
      DO 15 M = 1,NC
   15 A(L,M) = A(L,M) - A(I,M)*FAC
   20 CONTINUE
   25 CONTINUE
  301 FORMAT (1H /(20F6.2))
C
C CHECK FOR INCONSISTENCY AND DEPENDENCE
      NDF = 0
      DO 65 I = 1,NR
      J = 0
   45 J = J + 1
      IF (J.GT.NC) GC TO 55
      IF (A(I,J).EQ.0.) GO TO 45
      IF (J.LE.NCC) GC TO 65
      NDF = NDF + 1
      GO TO 65
   55 NR = NR - 1
      DO 60 K = I,NR
      L = K + 1
      DO 60 M = 1,NC
   60 A(K,M) = A(L,M)
   65 CONTINUE
      IF (NDF.EQ.0) GC TO 66
      WRITE (6,50) NDF
   50 FORMAT (1H-,28HTHE STRUCTURE IS A MECHANISM/30HNUMBER OF DEGREES OF
     *F FREEDOM =,I3)
   66 CONTINUE
C
C ISOLATE REDUNDANCIES
      NRD = 0
      DO 75 J = 1,NCC
      KOUNT = 0
      DO 70 I = 1,NR
   70 IF (A(I,J).NE.0.) KOUNT = KOUNT + 1
      IF (KOUNT.LE.1) GO TO 75
      NRD = NRD + 1
      MRED(NRD) = J
   75 CONTINUE
C
C REDUNDANCIES ... CHECK, OUTPUT
      I = NCC - NR
      IF (I.NE.NRD) WRITE (6,80)
   80 FORMAT (1H-,22H(COLS - ROWS) .NE. NRD)
      IF (NRD) 85,85,95
   85 WRITE (6,90)
   90 FORMAT (1H0,32HSTRUCTURE IS ALREADY DETERMINATE)
      GO TO 105
   95 WRITE (6,100) NRD,(MRED(I),I = 1,NRD)
  100 FORMAT (1H0,13HSTRUCTURE IS ,I2,16H TIMES REDUNDANT/
     * 22H REDUNDANT MEMBERS ARE,40I3)
  105 CONTINUE
C
C MOVE ALL REDUNDANT COLUMNS TO RHS OF MATRIX A
      IF (NRD.LE.0) GC TO 135
      DO 110 K = 1,NRD
  110 KNUT(K) = 0
      DO 130 I = 1,NRD
      IF (MRED(I).GT.NR) GO TO 130
      J = NR
```

```
  115 J = J + 1
      K = 0
  120 K = K + 1
      IF (MRED(K).EQ.J) GO TO 115
      IF (K.LT.NRD) GO TO 120
      K = MRED(I)
      DO 125 L = 1,NR
      BLOC = A(L,J)
      A(L,J) = A(L,K)
      A(L,K) = BLOC
  125 CONTINUE
      KNUT(I) = MRED(I)
      MRED(I)= J
  130 CONTINUE
  135 CONTINUE
C
C FORM UNIT MATRIX IN A(NR,NR)
      DO 155 LOO = 1,2
      DO 140 I = 1,NR
      DO 140 J = 1,NR
      IF (A(I,J).EQ.0.) GO TO 140
      MOP(I) = J
  140 CONTINUE
      DO 155 I = 1,NR
      IF (MOP(I).EQ.I) GO TO 155
      DO 145 K = I,NR
      IF (MOP(K).EQ.I) L=K
  145 CONTINUE
      DO 150 J = 1,NC
      BLOC = A(I,J)
      A(I,J) = A(L,J)
      A(L,J) = BLOC
  150 CONTINUE
      MOP(L) = MOP(I)
  155 CONTINUE
C
C INCREASE A TO GET COMPLETE INTERNAL LOAD SYSTEM
      K = NR + 1
      IF (NRD.LE.0) GO TO 170
      DO 165 I = K,NCC
      DO 160 J = K,NC
  160 A(I,J) = 0.
  165 A(I,I) = -1.
  170 CONTINUE
C
C BACK TO ORIGINAL ORDER OF ELEMENT FORCES
      IF (NRD.EQ.0) GO TO 174
      DO 173 I = 1,NRD
      IF (KNUT(I).EQ.0) GO TO 173
      J = KNUT(I)
      M = MRED(I)
      MRED(I) = J
      DO 172 L = K,NC
      BLOC = A(J,L)
      A(J,L) = A(M,L)
      A(M,L) = BLOC
  172 CONTINUE
  173 CONTINUE
  174 CONTINUE
      RETURN
```

```
      END


$IBFTC MANYLD  NODECK
      SUBROUTINE MANYLD (A,Q,OBJ,SENS,ACT,RHS,NTRS,NTIV,NTMBS,NTCA,
     * NTLDS,NMBS,NRD,NIV,NRS,NLDS,NRC)
C
      REAL A(NTMBS,NTCA),Q(NTMBS,NTLDS),OBJ(NTIV),SENS(NTRS,NTIV),ACT(NT
     * IV),RHS(NTRS)
C
      NIV = NMBS + NLDS*NRD
      NRS = NMBS*2*NLDS
      NC = NMBS + 1
      NCP = NMBS + NRC
      DO 10 I = 1,NMBS
      OBJ(I) = -OBJ(I)
   10 ACT(I) = -1.
      IF(NRD.EQ.0) GO TO 20
      DO 15 I = NC,NIV
      OBJ(I) = 0.
   15 ACT(I) = 1.
   20 CONTINUE
C
C SET UP SENSITIVITIES
      DO 30 I = 1,NRS
      DO 30 J = 1,NMBS
   30 SENS(I,J) = 0.
      DO 31 I = 1,NLDS
      DO 31 J = 1,NMBS
      K = 2*NMBS*(I - 1) + J
      L = K + NMBS
      SENS(K,J) = 1.
   31 SENS(L,J) = 1.
      IF (NRD.EQ.0) GO TO 40
      DO 32 I = 1,NRS
      DO 32 J = NC,NIV
   32 SENS(I,J) = 0.
      DO 35 I = 1,NLDS
      DO 35 J = 1,NMBS
      L = 2*NMBS*(I - 1) + J
      M = L + NMBS
      DO 35 K = 1,NRD
      N = NRD*(I - 1) + K + NMBS
      II = NRC + K
      SENS(L,N) = -A(J,II)
   35 SENS(M,N) = A(J,II)
   40 CONTINUE
C
C SELECT CRITICAL LOAD
      DO 55 I = 1,NMBS
      M = NMBS + I
      DO 55 K = 1,NLDS
      BLOC = 0.
      DO 45 J = 1,NRC
      L = NMBS + J
   45 BLOC = BLOC + A(I,L)*Q(J,K)
      M = 2*NMBS*(K - 1) + I
      N = M + NMBS
```

```
      RHS(M) = BLOC
      RHS(N) = -BLOC
   55  CONTINUE
C
C OUTPUT THE OPTIMISATION SUBROUTINE ARGUMENTS
      WRITE (6,601)NMBS,NIV
  601 FORMAT (1H , 5HNMBS-,I4/6H NIV- ,I4)
      WRITE (6,600) ((Q(I,J),I = 1,NRC),J = 1,NLDS)
  600 FORMAT (1H0,13HLOAD MATRIX -/(20F6.2))
      WRITE(6,602)((SENS(I,J),I=1,NRS ),J=1,NIV )
  602 FORMAT(1H0,6HSENS -/(10E12.3))
      WRITE (6,603) (OBJ(I), I=1,NIV )
  603 FORMAT (1H0,5HOBJ -/(10E12.3))
      WRITE (6,604) (ACT(I), I=1,NIV )
  604 FORMAT (1H0,5HACT -/(10E12.3))
      WRITE (6,607) (RHS(I), I=1,NRS )
  607 FORMAT (1H0,5HRHS -/(10E12.3))
      RETURN
      END



$IBFTC DUALP2
      SUBROUTINE DUALP2(SENS,OBJ,ACT,         RHS,  Z,           IPRNT,
     * CHS,KUT,IDEG,NAME,MARK,NTRS,NTMBS,NRZ,NCZ,M,N)
C
C           LINEAR PROGRAMMING BY SOLUTION OF DUAL
C
C           OUTPUT INDEX 'IPRNT' FOR THIS SUBROUTINE
C            IPRNT = 0  NO OUTPUT
C            IPRNT = 1  OUTPUT IFAS,MOVE LIMITS
C            IPRNT = 2  OUTPUT IFAS,MOVE LIMITS,Z-ARRAYS ETC.
C            IPRNT = 3  AS FOR 2 PLUS D-ROW OUTPUT FROM KRSIMP
C
      REAL SENS(NTRS,NTMBS),OBJ(NTMBS),ACT(NTMBS),
     * RHS(NTRS),          Z(NRZ,NCZ),CHS(NTMBS)
      INTEGER         OPT1,KUT(NCZ),IDEG(NCZ),NAME(NCZ),MARK(NRZ)
C
C           CLEAR Z-ARRAY
      DO 100 I = 1,NRZ
      DO 100 J = 1,NCZ
  100 Z(I,J) = 0.
C           SET DUAL SENSITIVITIES, OBJECTIVE COEFFICIENTS
C           AND ACTIVITY CODES IN Z-ARRAY
      ND = 0
      DO 113 J = 1,M
  104 DO 105 I = 1,N
  105 Z(I,J) = -SENS(J,I)
      Z(N+1,J) = RHS(J)
  113 CONTINUE
C           SET DUAL L.H. AND R.H. SIDES AND
C           CONSTRAINT CODES IN Z-ARRAY
  121 NZ = M + ND
  123 DO 130 I = 1,N
      Z(I,NZ+3) = OBJ(I)
      IF (ACT(I)) 124,124,125
  124 Z(I,NZ + 1) = 2.
      GO TO 130
  125 Z(I,NZ+1) = 3.
```

```
  130 CONTINUE
C           MAKE R.H.SIDES OF DUAL CONSTRAINTS POSITIVE
      DO 135 I = 1,N
      CHS(I) = -1.
      IF (Z(I,NZ+3)) 132,135,135
  132 CHS(I) = 1.
      DO 133 J = 1,NZ
  133 Z(I,J) = -Z(I,J)
      Z(I,NZ+3) = -Z(I,NZ+3)
      IF (Z(I,NZ+1).EQ.3.) GO TO 135
      Z(I,NZ+1) = 1.
  135 CONTINUE
C           OUTPUT Z-ARRAY BEFORE CALLING KRANTE
      MZ = N
      MZ2 = N + 2
      NZ3 = NZ + 3
      IF (IPRNT.LT.2) GO TO 140
      WRITE (6,136)
  136 FORMAT(30H0Z-ARRAY BEFORE CALLING KRANTE/)
      DO 137 I = 1,MZ2
  137 WRITE (6,138) (Z(I,J),J=1,NZ3)
  138 FORMAT(1H ,10E12.3)
  140 CONTINUE
      CALL KRANTE (Z,MZ,NZ,IFAS,KUT,NRZ,NCZ)
C           OUTPUT Z-ARRAY ETC. BEFORE CALLING KRSIMP
      MZ2 = MZ + 2
      NZ1 = NZ + 1
      IF (IPRNT.LT.2) GO TO 145
      WRITE (6,141)
  141 FORMAT(30H0Z-ARRAY BEFORE CALLING KRSIMP/)
      DO 142 I = 1,MZ2
  142 WRITE (6,138) (Z(I,J),J=1,NZ1)
      WRITE (6,144) IFAS,(KUT(J),J=1,NZ)
  144 FORMAT(7H0IFAS =,I4/12H0KUT ARRAY -/(1H ,24I5))
  145 CONTINUE
      CALL KRSIMP (Z,IFAS,MZ,NZ,NAME,IDEG,KUT,NRZ,NCZ,IPRNT,MARK)
      IF (IFAS.EQ.3) GO TO 162
C           OUTPUT Z-ARRAY ON RETURN FROM KRSIMP
      MZ1 = MZ + 1
      IF (IPRNT.LT.2) GO TO 150
      WRITE (6,146)
  146 FORMAT (30H0Z-ARRAY ON RETURN FROM KRSIMP/)
      DO 147 I = 1,MZ1
  147 WRITE (6,138) (Z(I,J),J=1,NZ1)
  150 CONTINUE
      IF (IPRNT.LT.1) GO TO 152
      WRITE (6,151) IFAS
  151 FORMAT (7H0IFAS =,I4)
  152 CONTINUE
C           SET VALUES OF PRIMAL VARIABLES IN FIRST ROW
C           OF Z-ARRAY, PRIMAL OBJECTIVE IN Z(2,1)
      DO 154 J = 1,MZ
  154 Z(1,J) = CHS(J) * Z(MZ1,J)
      Z(2,1) = -Z(2,1)
  160 CONTINUE
      RETURN
  162 Z(2,1) = 0.0
      RETURN
      END
```

```
$IBFTC FINAL   NODECK
      SUBROUTINE FINAL (Z,NRZ,NCZ,NMBS,NIV)
      REAL Z(NRZ,NCZ)
C
C OUTPUT OPTIMUM AREAS AND VCLUNE
      WRITE (6,10)
   10 FORMAT (1H-,50H**************************************************)
      WRITE (6,15) (Z(1,I),I = 1,NMBS)
   15 FORMAT(1H-,22HOPTIMUM MEMBER AREAS -/(6E12.4))
      J = NMBS + 1
      WRITE (6,20) (Z(1,I),I = J,NIV)
   20 FORMAT (1H-,22HVALUES OF REDUNDANTS -/(10E12.4))
      WRITE (6,25) Z(2,1)
   25 FORMAT (1H-, 19HSTRUCTURAL VOLUME  ,1PE12.4)
      RETURN
      END
```

APPENDIX D

Appendix D presents the listings of the six subroutines of the program SELF-WEIGHT PLASTIC DESIGN. This is a program to determine "efficient" weight designs of two- or three-dimensional pin-jointed trusses, with ideal elastic-plastic member load-deformation behaviours, for one or several loading cases, taking self-weight into account.

The program is dimensioned:

maximum number of joints $j = 16$
maximum number of members $m = 50$
maximum number of redundant members $r = 10$
maximum number of loading cases $c = 10$
(Note that in equations 5.9 and 5.10 maximum number of variables $= 60$).

The relevant matrices of the program are:

COORD is the matrix of joint coordinates
RELS is the matrix of support restraints
A is the augmented matrix $[C \vdots I]$
B is the force transformation matrix B_1
Q is the joint load matrix P
OBJ is the augmented member length matrix $[\ell^T \vdots 0^T]$
Z is the Simplex tableau
X is the vector of optimum area ratios a^*
R is the vector of optimum redundant force ratios r
QU is the matrix of the member force envelope $[R_{max} \vdots R_{min}]$
BO is the force transformation matrix B_o
VOL is the member volume matrices $[v_i^* \vdots v_{i+1}^*]$
MRED is the vector of redundant member numbers
ENDS is the matrix of the member incidence joint numbers
ALF is the compression coefficient α

```
$JOB             357206, MARKS
$TIME            60
$*               SELF-WEIGHT PLASTIC DESIGN
$IBJOB           MAP
$IBFTC SELFWT    NODECK
C PROGRAM FOR DESIGNING SPACE TRUSSES PLASTICALLY, INCLUDING SELF-WEIGHT AND
C  SEVERAL LOAD CONDITIONS
C  THE PROGRAM CONSISTS OF SIX SUBROUTINES -
C       SELFWT - VARIABLE DIMENSIONS AND SELF-WEIGHT ITERATIONS
C       PLOPT5 - FORMS MATHEMATICAL MODEL
C       RANK 2 - PROCESSES AND REARRANGES THE AUGMENTED MATRIX A
C       SAG    - SELF-WEIGHT INCREMENTS IN, FORMS L. P. PROBLEM
C       LIMFRA - FORMS THE TABLEAU AND SOLVES THE L.P. PROBLEM (DUAL SIMPLEX)
C       ULTIM - OUTPUTS THE RESULTS
C
      REAL COORD(16,3),RELS(16,3),A(50,100),B(50,10),Q(50,10),OBJ(60),
     * Z(110,70),X(160),R(10),AM(50),QU(50,2),BD(50,50),VOL(50,2)
      INTEGER MRED(10),MOP(50),KNUT(10),IZ(160),IXXP(60),IXP(110)    ,
     * ENDS(50,2)
      EQUIVALENCE (Z,A)
C
      DO 170 LOOK = 1,20
      CALL TIME
      CALL PLOPT5 (COORD,RELS,A,Q,OBJ,ENDS,16,50,10,100,60,NRC,NMBS,
     * NLDS,JF,NJS,NSJS)
      CALL RANK 2 (A,MRED,KNUT,MOP,50,100,10,NRC,NMBS,NRD)
      KOUNT = 1
      DO 22 I = 1,NMBS
      DO 22 J = 1,NRC
      K = J + NMBS
   22 BD(I,J) = A(I,K)
      WRITE (6,56) ((BD(I,J),J = 1,NRC),I = 1,NMBS)
   56 FORMAT (1H /(20F6.2))
      IF (NRD.EQ.0) GO TO 45
      DO 40 I = 1,NMBS
      DO 40 J = 1,NRD
      K = NMBS + 1 - J
   40 B(I,J) = A(I,K)
   45 CONTINUE
   10 CALL SAG (  Q,RELS,VOL,ENDS,OBJ,B,QU,BD,50,10,60,10,100,16,NMBS,
     * NRD,NLDS,JF,NJS,NSJS,ALF,KOUNT)
      CALL LDMFRE (B,QU,OBJ,Z,X,Y,R,IZ,IXXP,IXP,50,60,110,70,10,160,
     * NMBS,NRD,ALF)
      DO 11 I = 1,NMBS
   11 VOL(I,2) = 0.
      IF (KOUNT.LT.2) GO TO 13
      DO 12 I = 1,NMBS
   12 VOL(I,2) = VOL(I,1)
   13 CONTINUE
      DO 15 I = 1,NMBS
   15 VOL(I,1)=OBJ(I)*X(I)/10.
      XXX = Y
      IF (KOUNT.LT.2) GO TO 25
      IF (XXX.LT.YYY) GO TO 25
      IF ((XXX - YYY).LT..01) GO TO 30
      IF (KOUNT.GT.20) GO TO 30
   25 YYY = XXX
      CALL ULTIM (X,Y,R,AM,50,10,60,NMBS,NRD,KOUNT)
      KOUNT = KOUNT + 1
      GO TO 10
```

```
   30 CALL ULTIM (X,Y,R,AM,50,10,60,NMBS,NRO,KOUNT)
      CALL TIME (NM,NS,NSS)
C
      WRITE (6,169) NM,NS,NSS
  169 FORMAT (1H-,32HTIME TAKEN FOR ABOVE STRUCTURE -/20X,I4,5H MINS,I6,
     * 5H SECS,I6,6H SSECS)
  170 CONTINUE
      CALL EXIT
      END



$IBFTC PLOPT5
      SUBROUTINE PLOPT5 (COORD,RELS,A,Q,OBJ,ENDS,NTJS,NTMBS,NTLDS,NTCA,
     * NTIV,NRC,NMBS,NLDS,JF,NJS,NSJS)
      REAL COORD(NTJS,3),RELS(NTJS,3),A(NTMBS,NTCA),TYPE(2),XYZ(3),
     * PRNAME(13),DRNCOS(3),CJ(3),LENGTH,Q(NTMBS,NTLDS)
      REAL OBJ(NTIV)
      INTEGER ENDS(NTMBS,2)
      DATA TYPE/6H PLANE, 6H SPACE/, XYZ/1HX, 1HY, 1HZ/, FIN/6HFINISH/
C
C READ HEADER CARD
      READ (5,10) PRNAME
   10 FORMAT (13A6)
      IF (PRNAME(1).EQ.FIN) CALL EXIT
      WRITE (6,15) PRNAME
   15 FORMAT (1H1, 13A6)
C
C READ PROBLEM PARAMETERS -
C   JF = 2 FOR TWO-DIMENSIONAL TRUSS
C   JF = 3 FOR THREE-DIMENSIONAL TRUSS
C  NJS = NUMBER OF JOINTS IN TRUSS
C NSJS = NUMBER OF SUPPORT JOINTS IN TRUSS
C NMBS = NUMBER OF MEMBERS IN TRUSS
      WRITE (6,20)
   20 FORMAT (1H0, 10X, 6HDATA -)
      READ(5,25) JF, NJS, NSJS, NMBS
   25 FORMAT (20I4)
      NFJS = NJS - NSJS
C
C OUTPUT PROBLEM PARAMETERS
      WRITE (6,30) TYPE(JF - 1)
   30 FORMAT (1H0,20X,20HTYPE OF STRUCTURE  -,A6,29H TRUSS WITH THE CROSS-S
     *S-SECTION/42X,27HAREA OF EACH MEMBER VARYING)
      WRITE (6,35) NJS, NSJS, NMBS
   35 FORMAT (1H0,20X,20HNUMBER OF JOINTS   =,I4/31X,10HSUPPORTS =,I4/3
     *1X,10HMEMBERS  =,I4)
C
C READ JOINT NUMBERS AND COORDINATES -
C FREE JOINTS MUST BE NUMBERED FIRST, THEN SUPPORTS
C IF RELS(N,1) = 1. ... RESTRAINT AT N IN X-DIRN.
C IF RELS(N,2) = 1. ... RESTAINT AT N IN Y-DIRN.
C IF RELS(N,3) = 1. ... RESTAINT AT N IN Z-DIRN.
      DO 39 I = 1,NJS
      DO 39 J = 1,JF
   39 RELS(I,J) = 0.
      DO 40 I = 1,NJS
      READ (5,45) N, (COORD(N,J),J = 1,JF)
   45 FORMAT (I4, 3F8.4)
```

```
      IF (N.LE.NFJS) GO TO 40
      READ (5,46) (RELS(N,J),J = 1,JF)
   46 FORMAT (3F4.1)
   40 CONTINUE
C
C OUTPUT JOINT NUMBERS AND COORDINATES - FIRST FOR FREE JOINT AND THEN
C SUPPORT JOINTS
      WRITE (6,50) (XYZ(I),I=1,JF)
   50 FORMAT (1H0,20X,20HJOINT COORDINATES  -//28X,5HJOINT,10X,3(A1,11X))
     *)
      WRITE (6,55)
   55 FORMAT (23X,4HFREE)
      DO 60 I = 1,NJS
      WRITE (6,65) I, (COORD(I,J),J=1,JF)
   60 IF (I.EQ.NFJS) WRITE (6,70)
   65 FORMAT(1H ,I31,2X,6F12.4)
   70 FORMAT (23X,7HSUPPORT)
      NFJSP = NFJS + 1
      WRITE (6,47) ((RELS(I,J),J = 1,JF),I = NFJSP,NJS)
   47 FORMAT (1H0,20X,6HRELS -/27X,(6F5.1))
      WRITE (6,75) (XYZ(I),I = 1,JF)
   75 FORMAT (1H0,20X,17HMEMBER DETAILS  -//31X,6HMEMBER,6X,5HSTART,7X,
     * 3HEND,5X,6HLENGTH,3(3X,7HDRNCOS-,A1))
C
C LOOP ENTERED FOR ALL MEMBERS
      NRC = JF*NJS
      DO 79 I = 1,NRC
      DO 79 J = 1,NMBS
   79 A(I,J) = 0.
      NC = NMBS + 1
C
C
C READ MEMBER INCIDENCES AND PROPERTIES
      DO 120 MB = 1,NMBS
      READ (5,80) M,JNTST,JNTEND
   80 FORMAT (3I4, F12.4)
      IF (                       (JNTEND.LE.NJS).AND.(JNTST.LT.JNTEND)) GO
     * TO 90
      WRITE (6,85) M
   85 FORMAT (1H-,20H INCIDENCE OF MEMBER,I4,39H INCORRECTLY SPECIFIED.
     * JOB TERMINATED)
      CALL EXIT
   90 CONTINUE
      ENDS(M,1) = JNTST
      ENDS(M,2) = JNTEND
C CALCULATE MEMBER LENGTHS
      TLGTH = 0.
      DO 95 I = 1,JF
   95 TLGTH = TLGTH + (COORD(JNTEND,I) - COORD(JNTST,I))**2
      LENGTH = SQRT(TLGTH)
      OBJ(M) = LENGTH
C
C CALCULATE MEMBER DIRECTION COSINES
      DO 100 I = 1,JF
  100 DRNCOS(I) = (COORD(JNTEND,I) - COORD(JNTST,I))/LENGTH
C
C OUTPUT MEMBER INCIDENCES AND PROPERTIES
      WRITE (6,105) M,JNTST,JNTEND,LENGTH,(DRNCOS(I),I = 1,JF)
  105 FORMAT (1H ,23X,3I11,F12.4,3E11.3)
C
```

```
C ADD MEMBER OIRECTICN COSINES INTO THE CONNEXION MATRIX
      LROCS = JF*JNTST
      IROCS = LROCS - JF + 1
      DO 110 I = IROCS, LROCS
      J = I - IROCS + 1
  110 A(I,M) = -ORNCOS(J)
      LROCE = JF*JNTEND
      IROCE = LROCE - JF +1
      DO 115 I = IROCE, LROCE
      J = I - IROCE + 1
  115 A(I,M) = ORNCOS(J)
  120 CONTINUE
C
C
C READ SIZES OF LOAD COMPONENT AT EACH JOINT
C MUST NUMBER IN ORDER X, Y, Z, AT EACH JOINT
C NLJTS = NUMBER OF LOADED JOINTS
C NLDS = NUMBER OF LOADING CASES
      READ (5,140) NLDS
      DO 154 LOAD = 1,NLDS
      READ (5,140) NLJTS
  140 FORMAT (16I4)
      DO 145 I = 1,NRC
  145 Q(I,LOAD) = 0.
      DO 155 NO = 1,NLJTS
      READ (5,150) N, (QJ(I), I = 1,JF)
  150 FORMAT (I4,9F8.4)
      DO 155 I = 1,JF
      J = JF*(N-1) + I
  155 Q(J,LOAD) = CJ(I)
  154 CONTINUE
C
C TO CORRECT MATRICES C AND Q FOR SUPPORTS
      NFJSP = NFJS + 1
      M = 0
      DO 157 I = NFJSP, NJS
      DO 157 J = 1, JF
      IF (RELS(I,J).NE.1.) GO TO 157
      K = JF*(I - 1) + J - M
      NRC = NRC - 1
      DO 156 L = K, NRC
      N = L + 1
      DO 156 II = 1, NMBS
      A(L,II) = A(N,II)
  156 CONTINUE
      M = M + 1
  157 CONTINUE
C
C OUTPUT CONNEXION MATRIX
      WRITE (6,125)
  125 FORMAT ( 1H-, 18HCONNEXION MATRIX C/)
      DO 130 I = 1,NRC
  130 WRITE (6,135) (A(I,J),J = 1,NMBS)
  135 FORMAT (1H , 10E12.3)
  170 CONTINUE
      RETURN
      END
```

```
$IBFTC RANK 2
      SUBROUTINE RANK 2 (A,MRED,KNUT,MOP,NTMBS,NTCA,NTIV,NR,NCC,NRD)
      REAL A(NTMBS,NTCA)
      INTEGER MRED(NTIV),KNUT(NTIV),MOP(NTMBS)
C
C SET UP UNIT SUBMATRIX
      NCP = NCC + 1
      NC = NCC + NR
      DO 4 I = 1,NR
      DO 3 J = NCP,NC
    3 A(I,J) = 0.
      K = I + NCC
    4 A(I,K) = 1.
C
C JORDAN ELIMINATION
      DO 25 I = 1,NR
      DEN = 0.
      DO 5 J = 1,NCC
      AB = ABS(A(I,J))
      DN = ABS(DEN)
      IF (AB.GT.DN) JJ = J
      IF (AB.GT.DN) DEN = A(I,J)
    5 CONTINUE
      IF (DEN.EQ.0.) GO TO 25
      DO 10 K = 1,NC
   10 A(I,K) = A(I,K)/DEN
      DO 20 L = 1,NR
      IF (L.EQ.I) GO TO 20
      FAC = A(L,JJ)
      DO 15 M = 1,NC
   15 A(L,M) = A(L,M) - A(I,M)*FAC
   20 CONTINUE
   25 CONTINUE
  301 FORMAT (1H /(20F6.2))
C
C CHECK FOR INCONSISTENCY AND DEPENDENCE
      NDF = 0
      DO 65 I = 1,NR
      J = 0
   45 J = J + 1
      IF (J.GT.NC) GO TO 55
      IF (A(I,J).EQ.0.) GO TO 45
      IF (J.LE.NCC) GC TO 65
      NDF = NDF + 1
      GO TO 65
   55 NR = NR - 1
      DO 60 K = I,NR
      L = K + 1
      DO 60 M = 1,NC
   60 A(K,M) = A(L,M)
   65 CONTINUE
      IF (NDF.EQ.0) GO TO 66
      WRITE (6,50) NDF
   50 FORMAT (1H-,28HTHF STRUCTURE IS A MECHANISM/30HNUMBER OF DEGREES OF
     *F FREEDOM =,I3)
   66 CONTINUE
C
C ISOLATE REDUNDANCIES
      NRD = 0
```

```
      DO 75 J = 1,NCC
      KOUNT = 0
      DO 70 I = 1,NR
   70 IF (A(I,J).NE.0.) KOUNT = KOUNT + 1
      IF (KOUNT.LE.1) GO TO 75
      NRD = NRD + 1
      MRED(NRD) = J
   75 CONTINUE
C
C REDUNDANCIES ... CHECK, OUTPUT
      I = NCC - NR
      IF (I.NE.NRD) WRITE (6,80)
   80 FORMAT (1H-,22H(COLS - ROWS) .NE. NRD)
      IF (NRD) 85,85,95
   85 WRITE (6,90)
   90 FORMAT (1H0,32HSTRUCTURE IS ALREADY DETERMINATE)
      GO TO 105
   95 WRITE (6,100) NRD,(MRED(I),I = 1,NRD)
  100 FORMAT (1H0,13HSTRUCTURE IS ,I2,16H TIMES REDUNDANT/
     * 22H REDUNDANT MEMBERS ARE,40I3)
  105 CONTINUE
C
C MOVE ALL REDUNDANT COLUMNS TO RHS OF MATRIX A
      IF (NRD.LE.0) GO TO 135
      DO 110 K = 1,NRD
  110 KNUT(K) = 0
      DO 130 I = 1,NRD
      IF (MRED(I).GT.NR) GO TO 130
      J = NR
  115 J = J + 1
      K = 0
  120 K = K + 1
      IF (MRED(K).EQ.J) GO TO 115
      IF (K.LT.NRD) GC TO 120
      K = MRED(I)
      DO 125 L = 1,NR
      BLOC = A(L,J)
      A(L,J) = A(L,K)
      A(L,K) = BLOC
  125 CONTINUE
      KNUT(I) = MRED(I)
      MRED(I)= J
  130 CONTINUE
  135 CONTINUE
C
C FORM UNIT MATRIX IN A(NR,NR)
      DO 155 LOO = 1,2
      DO 140 I = 1,NR
      DO 140 J = 1,NR
      IF (A(I,J).EQ.0.) GO TO 140
      MOP(I) = J
  140 CONTINUE
      DO 155 I = 1,NR
      IF (MOP(I).EQ.I) GO TO 155
      DO 145 K = I,NR
      IF (MOP(K).EQ.I) L=K
  145 CONTINUE
      DO 150 J = 1,NC
      BLOC = A(I,J)
      A(I,J) = A(L,J)
```

```
      A(L,J) = BLOC
  150 CONTINUE
      MOP(L) = MOP(I)
  155 CONTINUE
C
C INCREASE A TO GET COMPLETE INTERNAL LOAD SYSTEM
      K = NR + 1
      IF (NRD.LE.0) GO TO 170
      DO 165 I = K,NCC
      DO 160 J = K,NC
  160 A(I,J) = 0.
  165 A(I,I) = -1.
  170 CONTINUE
C
C BACK TO ORIGINAL ORDER OF ELEMENT FORCES
      IF (NRD.EQ.0) GC TO 174
      DO 173 I = 1,NRD
      IF (KNUT(I).EQ.0) GO TO 173
      J = KNUT(I)
      M = MRED(I)
      MRED(I) = J
      DO 172 L = K,NC
      BLOC = A(J,L)
      A(J,L) = A(M,L)
      A(M,L) = BLOC
  172 CONTINUE
  173 CONTINUE
  174 CONTINUE
      RETURN
      END


$IBFTC SAG     NODECK
      SUBROUTINE SAG (  Q,RELS,VOL,ENDS,OBJ,B,QU,BD,NTMBS,NTRO,NTIV,
     * NTLDS,NTCA,NTJS,NMBS,NRD,NLDS,JF,NJS,NSJS,ALF,KOUNT)
      REAL              Q(NTMBS,NTLDS),RELS(NTJS,3),VOL(NTMBS,2),
     * OBJ(NTIV),B(NTMBS,NTRO),QU(NTMBS,2),BD(NTMBS,NTMBS)
      INTEGER ENDS(NTMBS,2)
C
C CONSTANTS
      NFJS = NJS - NSJS
      NFJSP = NFJS + 1
C
C INCREASE LOAD MATRIX TO INCLUDE SUPPORTS
      IF (KOUNT.LT.2) GC TO 10
      DO 4 I = NFJSP,NJS
      DO 4 J = 1,JF
      IF (RELS(I,J).NE.1.) GO TO 4
      K = JF*(I - 1) + J
      L = K + 1
      NRC = NRC + 1
      DO 2 N = L,NRC
      IJ = NRC + L - N
      IK = IJ - 1
      DO 2 II = 1,NLDS
    2 Q(IJ,II) = Q(IK,II)
    4 CONTINUE
C
```

```
C ADD SELF-WEIGHT TO LOAD MATRIX
      DO 5 I = 1,NMBS
      J = JF*ENDS(I,1)
      K = JF*ENDS(I,2)
      DO 5 L = 1,NLDS
      Q(J,L) = Q(J,L) - VOL(I,1)/2. + VOL(I,2)/2.
    5 Q(K,L) = Q(K,L) - VOL(I,1)/2. + VOL(I,2)/2.
C
C CORRECT LOAD MATRIX FOR SUPPORTS
   10 NRC = JF*NJS
      M = 0
      DO 20 I = NFJSP,NJS
      DO 20 J = 1,JF
      IF (RELS(I,J).NE.1.) GO TO 20
      K = JF*(I - 1) + J - M
      NRC = NRC - 1
      DO 15 L = K,NRC
      N = L + 1
      DO 15 II = 1,NLDS
   15 Q(L,II) = Q(N,II)
      M = M + 1
   20 CONTINUE
C
C SELECT CRITICAL LOAD PATTERN
      NIV = NMBS + NRD
      NCP = NMBS + 1
      NC = NMBS + NRC
      DO 35 I = 1,NMBS
      DO 35 K = 1,NLDS
      BLOC = 0.
      DO 25 J = 1,NRC
   25 BLOC = BLOC + BO(I,J)*Q(J,K)
      IF (K.GT.1) GO TO 30
      QU(I,1) = BLOC
      QU(I,2) = BLOC
      GO TO 35
   30 IF (BLOC.GT.QU(I,1)) QU(I,1) = BLOC
      IF (BLOC.LT.QU(I,2)) QU(I,2) = BLOC
   35 CONTINUE
C
C CALCULATE OTHER COEFFICIENTS
      WRITE (6,11)
   11 FORMAT (1H-,50H**************************************************)
      IF (NRD.EQ.0) GO TO 50
      DO 45 I = 1,NRD
      J = NMBS + I
   45 OBJ(J) = 0.
      IF (KOUNT.GT.1) GO TO 50
      WRITE (6,602) ((B(I,J),I = 1,NMBS),J = 1,NRD)
  602 FORMAT(1H0,6HSENS -/(10E12.3))
   50 CONTINUE
C
      ALF = 1.
   55 FORMAT (F12.4)
      IF (KOUNT.GT.1) GO TO 65
      WRITE (6,60) ALF
   60 FORMAT (1H0,26HCOMPRESSION COEFFICIENT IS,F12.4)
C
      WRITE (6,601)NMBS,NIV
  601 FORMAT (1H , 5HNMBS-,I4/6H NIV- ,I4)
```

```
      WRITE (6,603) (OBJ(I), I=1,NIV )
  603 FORMAT (1H0,5HOBJ -/(10E12.3))
   65 CONTINUE
      WRITE (6,607)((QU(I,J),I = 1,NMBS),J = 1,2)
  607 FORMAT (1H0,5HRHS -/(10E12.3))
C
C WRITE LOAD MATRIX Q
      WRITE (6,160)
  160 FORMAT (1H0,16HLOAD MATRIX Q  -)
      DO 163 J = 1,NLDS
  163 WRITE (6,165) (Q(I,J),I = 1,NRC)
  165 FORMAT(1H ,18F6.3)
      RETURN
      END
```

```
$IBFTC LOMFRE
      SUBROUTINE LOMFRE (B,Q,OBJ,Z,X,Y,R,IZ,IXXP,IXP,NTMBS,NTIV,NRZ,
     * NCZ,NTRD,NTN1,NMBS,NRD,ALF)
      REAL B(NTMBS,NTRD),Q(NTMBS,2),OBJ(NTIV),Z(NRZ,NCZ),X(NTN1),R(NTRD)
      INTEGER IZ(NTN1),IXXP(NTIV),IXP(NRZ)
C
C SET UP TABLEAU
C
C CALCULATE SIZE LIMITS
      NP = NMBS + NRD
      NP1 = NP + 1
      ND = NMBS*2
      M1 = ND + 1
      N1 = NP + ND
      IH = NMBS + 1
C
C CLEAR 'Z' TABLEAU
      DO 5 I = 1,M1
      DO 5 J = 1,NP1
    5 Z(I,J) = 0.
C
C SENSITIVITIES, RHS, AND OBJ IN
      DO 20 I = 1,NMBS
      K = I*2
      J = K - 1
      Z(M1,I) = -OBJ(I)
      Z(J,I) = -1.
      Z(K,I) = -ALF
      JJ = NP + J
      KK = NP + K
      IXP(J) = JJ
      IXP(K) = KK
      IZ(JJ) = J
      IZ(KK) = K
      XX = -Q(I,1)
      YY = +Q(I,2)
      IF (NRD.EQ.0) GO TO 16
      DO 15 L = IH,NP
      M = L - NMBS
      Y = B(I,M)
      Z(J,L) = Y
      Z(K,L) = -Y
```

```
      XX = XX + Y*100.
   15 YY = YY - Y*100.
   16 Z(J,NP1) = XX
   20 Z(K,NP1) = YY
C
C POINTERS SET UP
      DO 25 J = 1,NP
      IXXP(J) = J
      IZ(J) = -J
   25 X(J) = 0.
      NUM = 1
C
C START ITERATIONS TO OBTAIN OPTIMUM TABLEAU
C
   30 CONTINUE
      DO 35 I = 1,ND
      J = IXP(I)
   35 X(J) = Z(I,NP1)
C
C CALCULATE PIVOT ROW
C
      Y = 0.
      DO 65 I = 1,ND
      A1 = ABS(Z(I,NP1))
      IF (A1.GT.Y) Y = A1
   65 CONTINUE
C
      XX = 0.
      IXX = 0
      Y = 1./Y
C
      DO 75 I = 1,ND
      A1 = Z(I,NP1)
      A1Y = A1*Y
      AA1 = ABS(A1Y)
      IF ((AA1.GT.1.E-6).AND.(A1.LT.XX)) GO TO 70
      GO TO 75
   70 XX = A1
      LL = I
      IXX = 1
   75 CONTINUE
C
C IS PIVOTING COMPLETE
C
      IF (IXX.EQ.0) GO TO 200
C
C CACLULATE PIVOT COLUMN
C
      Y = 0 .
      DO 80 J = 1,NP
      A1 = ABS(Z(LL,J))
      IF (A1.GT.Y) Y = A1
   80 CONTINUE
C
      XX = 1.E30
      Y = 1./Y
      KK = 0
C
      DO 85 J = 1,NP
      A11 = Z(LL,J)
```

```
      AY = A11*Y
      AAY = ABS(AY)
      IF ((AAY.LT.1.E-6).OR.(A11.GT.0.)) GO TO 85
      A1 = Z(M1,J)/A11
      IF (A1.GT.XX) GO TO 85
      XX = A1
      KK = J
   85 CONTINUE
C
C INFEASIBLE SOLUTION
C
      IF (KK.NE.0) GC TO 95
      WRITE (6,90)
   90 FORMAT (1H-,61HINFEASIBLE SOLUTION - NO NEGATIVE COEFFICIENT IN CRITICAL R
     *ITICAL ROW)
      RETURN
C
C PIVOTING
C
   95 A1 = 1./Z(LL,KK)
C
      DO 105 J = 1,NP1
      IF (J.EQ.KK) GO TO 105
      Z(LL,J) = A1*Z(LL,J)
      A11 = Z(LL,J)
      DO 100 I = 1,M1
      IF (I.EQ.LL) GC TO 100
      Z(I,J) = Z(I,J) - A11*Z(I,KK)
  100 CONTINUE
  105 CONTINUE
C
      DO 110 I = 1,M1
  110 Z(I,KK) = -A1*Z(I,KK)
C
      Z(LL,KK) = A1
C
C CHANGE POIMTERS ANC REITERATE
C
      NN = IXP(LL)
      MM = IXXP(KK)
      IZ(NN) = -KK
      IZ(MM) = LL
      IXP(LL) = MM
      IXXP(KK) = NN
      X(NN) = 0.
C
      NUM = NUM + 1
      GO TO 30
C
C CALCULATE CORRECT VARIABLE VALUES
C
  200 CONTINUE
      WRITE (6,36) NUM
   36 FORMAT (1H0,15HTABLEAU NUMBER ,I2/)
      DO 205 J = IH,NP
      K = J - NMBS
      R(K) = 0.
      IF (NRD.EQ.0) GO TO 205
      R(K) = X(J) - 100.
  205 CONTINUE
```

```
C
      Y = 0.
      DO 210 J = 1,NMBS
  210 Y = Y + OBJ(J) * X(J)
C
      RETURN
      END


$IBFTC ULTIM   NODECK
      SUBROUTINE ULTIM (X,Y,R,AM,NTMBS,NTRD,NTIV,NMBS,NRD,KOUNT)
      REAL X(NTIV),R(NTRD),AM(NTMBS)
C
C OUTPUT OPTIMUM AREAS AND VOLUME
      WRITE (6,10)
   10 FORMAT (1H0,50H--------------------------------------------------)
      WRITE (6,15) (X(I),I = 1,NMBS)
   15 FORMAT(1H0,22HOPTIMUM MEMBER AREAS -/(6E12.4))
      WRITE (6,20) (R(I),I = 1,NRD)
   20 FORMAT (1H0,22HVALUES OF REDUNDANTS -/(10E12.4))
      WRITE (6,25) Y
   25 FORMAT (1H0, 19HSTRUCTURAL VOLUME  ,1PE12.4)
   30 FORMAT (1H0,21HACTUAL AREAS NEEDED -/( 6E12.4))
      WRITE (6,35) KOUNT
   35 FORMAT (1H0,16HITERATION NUMBER,I3)
      RETURN
      END
```

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisor, Professor L.K. Stevens, who originally suggested this topic, and whose guidance and advice have been very encouraging.

The author would also like to thank Dr. W.T. O'Brien and Mr. D.W. Bennett who gave valuable help at the start of the work, and Dr. J.F. Brotchie and Dr. A.R. Toakley, both of the CSIRO Division of Building Research, who readily lent references and gave very useful advice during the project.

To Mrs. Jenny Parry he wishes to express his thanks for her assistance in preparing the report.

The work described in this report was supported by a Commonwealth Postgraduate Award.

BIBLIOGRAPHY

ARGYRIS, J.H. and KELSEY, S. (1960), "Energy Theorems and Structural Analysis", Butterworths, London.

BAKER, J. (1949), "Design of steel frames", Str. Engrg. 27, 397.

BAKER, J., HORNE, M.R. and HEYMAN, J. (1956) "The Steel Skeleton" II, Cambridge U.P., Cambridge.

BEALE, E.M.L. (1959) "On quadratic programming", U.S. Naval Research Logistics Quarterly, 6, 227.

BERTERO, V.V. (1965), discussion of PERRONE & SOTERIADES (1965), Proc. A.S.C.E., 91, EM 5, 218.

CHAN, H.S.Y. (1964), "Optimum structural design and linear programming", College of Aeronautics Report Aero. 175.

CHAN, H.S.Y. (1967), "Minimum volume design of frameworks and discs for alternate loading systems", Q. Appl. Math., 25, 490.

CHAN, H.S.Y. (1968), "Mathematical programming in optimal plastic design", Inter. J. Solids Struct., 4, (9), 885.

CHARLTON, T.M. (1950), "Some notes on analysis of redundant systems by means of conception of conservation of energy", J. Franklin Inst., 250, 543.

CHARLTON, T.M. (1951), discussion of SYMONDS & PRAGER (1950) (1), J. Appl. Mech., 18(1), 118.

CHARLTON, T.M. (1952), "Analysis of statically indeterminate structures by the complementary energy method" Engrg., 389.

CHARNES, A. and GREENBERG, H.J. (1951), "Plastic collapse and linear programming", Bull. of the Am. Math. Soc., 57(6), 480.

CHARNES, A., LEMKE, C.E., and ZIENKIEWICZ, O.C. (1959), "Virtual work, linear programming, and plastic limit analysis", Proc. Roy. Soc., A 251, 110.

DAVIES, J.M. (1967), "Collapse and shakedown loads of plane frames", Proc. A.S.C.E., 93, ST3.

DENKE, P.H. (1965), "A Computerised Static and Dynamic Structural Analysis System" III, S.A.E. Special Publication, SP-264.

DORN, W.S. (1960), "A duality theorem for convex programs", IBM J.R. & D., 4, 4.

DORN, W.S., GOMORY, R.E., and GREENBERG, H.J. (1964), "Automatic design of optimal structures", J. de Mecanique, 3, 1.

DORN, W.S. and GREENBERG, H.J. (1957), "Linear programming and plastic limit analysis of structures", Q. Appl. Math., 15, 155.

DRUCKER, D.C., GREENBERG, H.J., and PRAGER, W. (1951), "Safety factor of an elastic-plastic body in plane strain", J. Appl. Mech., 18, 371.

DRUCKER, D.C. and SHIELD, R.T. (1956), "Design for minimum weight", Proc. 9th Inter. Cong. Appl. Mech., B5, 212.

FINZI, L. (1957), "Unloading processes in elastic-plastic structures", Proc. 9th Inter. Cong. Appl. Mech.

FOULKES, J.D. (1953), "Minimum weight design and theory of plastic collapse", Q. Appl. Math., 10, 347.

FOULKES, J.D. (1954), "Minimum weight design of structural frames", Proc. Roy. Soc., A 223, 482.

FOULKES, J.D. (1955), "Linear programming and structural design", Proc. 2nd Symp. Linear Programming, 117.

GREENBERG, H.J. (1949), "Complementary minimum principles for elastic plastic material", Q. Appl. Math., 6, 85 (1949/50).

GREENBERG, H.J. and PRAGER, W. (1951), "Limit design of beams and frames", Proc. A.S.C.E., 77, 12, 1951; Trans. A.S.C.E., 117, 447, 1952.

GREGORY, M. (1962), discussion of HEYMAN (1961), Proc. I.C.E., 23, 300.

GURFINKEL, G. (1965), discussion of PERRONE & SOTERIADES (1965), Proc. A.S.C.E., 91, EM 4.

HAAR, A. and von KARMAN, Th. (1909), "Zur Theorie der Spannungs zustände in plastischen und sandartigen Medien" Göttingen Nochrichten, math-phys., 204.

HEYMAN, J. (1951) (1), "Limit design of space frames", J. Appl. Mech., 18(2), 157.

HEYMAN, J. (1951) (2),"Plastic design of beams and plane frames for minimum volume", Q. Appl. Math., 8, 373.

HEYMAN, J. (1953), "Plastic design of plane frames for minimum weight", Str. Engr., 31, 125.

HEYMAN, J. (1959) (1), "Automatic analysis of steel framed structures under fixed and varying loads", Proc. I.C.E., 12, 39.

HEYMAN, J. (1959) (2), "On the absolute minimum weight design of framed structures", Q. J. Mech. Appl. Math., 12, 314.

HEYMAN, J. (1961), "On the estimation of deflections in elastic-plastic framed structures", Proc. I.C.E., 19, 39.

HEYMAN, J. (1968), "Bending moment distribution in collapsing frames"from 'Engrg. Plastcty', ed. Leckie & Heyman C.U.P.

HEYMAN, J. and PRAGER, W. (1958), "Automatic minimum weight design of steel frames", J. Franklin Inst., 266,(5), 339.

HODGE, P.G. Jn. (1959),"Plastic Analysis of Structures", McGraw-Hill, N.Y.

HORNE, M.R. (1948), discussion of HRENNIKOFF (1948), Trans. A.S.C.E., 113, 213.

HORNE, M.R. (1950), "Fundamental properties in the plastic theory of structures", J.I.C.E., 34, 174.

HORNE, M.R. (1954), "A moment distribution method for analysis and design of structures by plastic methods", Proc. I.C.E., 3(3), 51.

HORNE, M.R. (1962), discussion of HEYMAN (1961), Proc. I.C.E., 23, 298.

HOSKIN, B.C. (1960), "Limit analysis, limit design, and linear programming", Aeronautical Research Lab., ARL/SM 274.

HRENNIKOFF, A. (1948), "Theory of inelastic bending with reference to limit design", Trans. A.S.C.E., 113, 213.

HRENNIKOFF, A. (1965), "Importance of strain hardening in plastic design", Proc. A.S.C.E., 91, ST4.

JENNINGS, A. and MAJID, K. (1965), "Elastic-plastic analysis by computer for framed structures to collapse," J.I.St.E., 43, 407.

KICHER, T.P. (1966), "Optimum design-minimum weight versus fully stressed", Proc. A.S.C.E., 92, ST6.

KNUDSEN, YOUNG, JOHNSON, BEEDLE, and WEISKOPF (1953), "Plastic strength and deflections of continuous beams", J. Weld., 32, 240.

KOOPMAN, D.C.A. and LANCE, R.H. (1965), "On linear programming and plastic limit analysis", J. Mech. Phys. Solids, 13, 77.

KORN, A. and GALAMBOS, T.V. (1968), "Behaviour of elastic-plastic frames", Proc. A.S.C.E., 94, ST5.

KÜNZI, H.P., TZSCHACHE, H.G., and ZEHNDER, C.A. (1968), "Numerical Methods of Mathematical Optimisation", Academic Press, N.Y.

LEE, S.L. (1958), "Conjugate force method and its application in elastic and plastic theory of structures", J. Franklin Inst., 266, 207.

LIND, N.C. (1965), "Analysis of deflections in elastic-plastic frames", Proc. A.S.C.E., 91, ST3.

LIVESLEY, R.K. (1956), "Automatic design of structural frames". Q. J. Mech. Appl. Math., 9, 257.

LIVESLEY, R.K. (1959), "Optimum design of structural frames for alternative systems of loading", Civ. Engrg & P.W.R., 54, 737.

LIVESLEY, R.K. (1964), "Matrix Methods of Structural Analysis", Pergammon, Oxford.

LIVESLEY, R.K. (1966), "Systematic method for the collapse analysis of plane frameworks", Inter. Symp. E.D.C. in S.E., Newcastle.

LIVESLEY, R.K. (1967), "Selection of redundant forces in structures...", Proc. Roy. Soc., A 301.

MARTIN, J.B. (1962), discussion of HEYMAN (1961), Proc. I.C.E., 23, 298.

MATHESON, J.A.L. (1959), "Hyperstatic Structures" I, Butterworths, London.

MAYEDA, R. and PRAGER, W. (1967), "Minimum weight design of beams for multiple loading", Inter. J. Solids Struct., 3, 1001.

MEGAREFS and HODGE, D.G. (1963), "Method for plastic design of frames", Proc. A.S.C.E., 89, ST1, 197.

MICHELL, A.G.M. (1904), "Limit of economy of material in frame structures", Phil. Mag. Series 7, 8, 589.

NEAL, B.G. (1950), "Plastic collapse and shakedown theorems for structures of strain hardening material", J. Aero. Scis., 17, 297.

NEAL, B.G. (1956), "Plastic Methods of Structural Analysis", Chapman & Hall, London.

NEAL, B.G. and SYMONDS, P.S. (1950), "Calculation of failure loads on plane frames under aribtrary loading", J.I.C.E., 35, 214.

NEAL, B.G. and SYMONDS, P.S. (1952), "Rapid calculation of the plastic collapse load for a framed structure", Proc. I.C.E. (3), 1, 58.

ODEN, J.T. (1967), "Linear and non-linear analysis by conjugate theory", Proc. A.S.C.E., 93, ST4.

PEARSON, C.E. (1958), "Structural design by high speed computing machines", A.S.C.E. Conf. Elect. Comp.

PERRONE, N. and SOTERIADES, M.C. (1965), "Calculating deflections of a partially collapsed frame", Proc. A.S.C.E., 91, EM1.

PRAGER, W. (1948), "Plastic flow theory versus plastic deformation theory", J. Appl. Phys., 19, 540.

PRAGER, W. (1956), "Minimum weight design of portal frames", Proc. A.S.C.E., 82, EM4.

PRAGER, W. (1959), "Introduction to Plasticity", Addison-Westley, Reading, Mass.

PRAGER, W. (1965), "Mathematical programming and theory of structures", J. S.I.A.M., 13, 312.

PRAGER, W. (1967), "Optimum plastic design of portal frame for alternating loads", J. Appl. Mech., 34, 772.

PRAGER, W. and SHIELD, R.T. (1967), "General theory of optimal plastic design", J. Appl. Mech., 34, 184.

REINSCHMIDT, K.F., CORNELL, C.A., and BROTCHIE, J.F. (1966), "Iterative design and structural optimisation", Proc. A.S.C.E., 92, ST6.

ROBINSON, J. (1966), "Structural Matrix Analysis for the Engineer", John Wiley, N.Y.

ROBINSON, J. (1968), "General note on the selection of redundants in structural analysis", Aeron. J., 72, 626.

SAATY, T.L. and BRAM, (1964), "Nonlinear Mathematics", McGraw-Hill, N.Y.

SAVE, M. and PRAGER, W. (1963), "Minimum weight design of beams subject to fixed and moving loads", J. Mech. Phys. Solids, 11, 252.

SHEU, C.Y. and PRAGER, W. (1968), "Recent developments in optimal structural design", Appl. Mech. Rev., 21, 10, Oct. 1968.

SHIELD, R.T. (1963), "Optimum design for multiple loading", J. Appl. Math. Phys. (Z.A.M.P.), 14, 38.

STEVENS, L.K. (1960), "Direct design by limiting deformations", Proc. I.C.E., 16, 235.

STEVENS, L.K. (1968), "Plastic design and trussed frames", from "Engrg. Plastcty.", ed. Leckie & Heyman, C.U.P.

SVED, G. (1954), "Minimum weight of certain redundant structures", Aust. J. Appl. Sci., 3, 1.

SYMONDS, P.S. and NEAL, B.G. (1951), "Recent progress in plastic methods of structural analysis", J. Franklin Inst., 254(2), 473.

SYMONDS, P.S. and NEAL, B.G. (1952), "Interpretation of failure loads in plastic theory of continuous beams and frames", J. Aero. Scis., 19, 95.

SYMONDS, P.S. and PRAGER, W. (1950) (1), "Elastic-plastic analysis of structures subjected to loads varying arbitrarily...", J. Appl. Mech., 17, 315.

SYMONDS, P.S. and PRAGER, W. (1950) (2), "Stress analysis in elastic-plastic structures", Proc. 3rd. Symp. Appl. Math.

THODANI, B.N. (1966), "Deflexion calculations in plastic analysis", Engrg. J., 49, April 1966.

TOAKLEY, A.R. (1967), "Optimum elastic-plastic design of rigid-jointed sway frames", Research Report 4, Dept. of Civ. Engrg., Manchester.

TOAKLEY, A.R. (1968), "Some computational aspects of optimum rigid-plastic design", Int. J. Mech. Sci., 10, 531.

TOCHER, J.L. and POPOV, E.P. (1962), "Incremental collapse analysis of rigid frames", Proc. 4th U.S. Natl. Cong. Appl. Mech.

VAN DEN BROEK, J.A. (1948), "Theory of Limit Design", John Wiley, N.Y.

WANG, C-K. (1966), "General computer program for limit analysis", Proc. A.S.C.E., 89, ST6.

WASHIZU, K. (1968), "Variational Methods in Elasticity and Plasticity", Pergammon Press, Oxford.

WESTERGAARD, H.M. (1942), "On the method of complementary energy", Trans. A.S.C.E., 107, 765.

WILLIAMS, D. (1938), "Relations between the energy theorems applicable in structural theory", Phil. Mag., WOLFE, P. (1959), "Simplex method for quadratic programming", Econometrica, 27, 3.

WRIGHT, J. and BATY, J. (1966), "Plastic analysis and design of rigid-plastic frameworks subjected to multi-load conditions", Inter. Symp. E.D.C. in S.E., Newcastle.

EXAMINER'S REPORT

"Optimisation and Plastic Analysis" - R. E. Marks

General Comments

A topic of substantial significance in structural design is tackled in the thesis, and a theoretically sound solution to various aspects has been given within rather sweeping simplifications of actual structural behaviour. The work parallels other current investigations where the potential of linear programming in optimisation of structures has been recognised.

The thesis revealed a real understanding of the problems tackled, but the presentation was such that this understanding was not easily communicated to the reader. Reasons for this are given in the following specific comments.

Specific Comments

(1) The Literature Survey revealed a knowledge of the significant work, including some references which had escaped this examiner's notice. However, it read almost like a bibliography with the barest of commentary, in no way contributing to the thesis which effectively begins with Chapter III. The commentary only has meaning to the initiated, already familiar with the jargon (which begins in Chapter I).

Terms such as "overcomplete collapse", "quadratic programming" and "minimum structural complementary energy" appear without warning or definition. The symbols used in Chapter II are not defined in the text until Chapter III.

This examiner was not happy with the problem of compression members in limit analysis and design. With regard to pure elastic buckling of struts (p.18), the author agrees with NEAL (1950): "Hence, as in ideal plastic behaviour, the axial deformation increases at constant load." This is quite wrong, as a closer study of the elastica will show. The idealisation of load-shortening as given by Figure IV can only cast serious doubt on the validity of any practical design based on it. A subsequent load-history analysis using a proven plastic plateau of strut behaviour would be necessary. Nevertheless, the simplification is justified in this thesis, where some initial answers are being sought.

(2) The theory is adequately presented but made more difficult to follow by the introduction of non-dimensional functions and arguments, which double the amount of notation needed with no real gain in presentation. To some extent computing problems have dominated the thesis. This examiner knows of one standard linear programming routine available on a not very large

computer in Melbourne at the time this work was done which will handle up to 1024 variables and which automatically selects an initial basic feasible solution. A further routine for larger problems is also available. The section on Automatic Selection of Redundants is nevertheless interesting. The rather drastic limitations in size of tableau came as a surprise. The effort to make the SELF-WEIGHT PLASTIC DESIGN routine work may not have been made, in view of its poor economy in test cases, if a larger capacity of linear program was recognised to be readily available. No precise statement of tableau restrictions is given in the thesis.

In the multiload design problem, no mention was made of the shakedown criterion, which also leads to a linear programming problem. A possible future development would be a study of optimum structures comparing the effect of shakedown and plastic collapse criteria on minimum weight.

(3) The example problems were not clearly presented. Although they are there to demonstrate the theory and the correctness of the computer programs, they should also help spell out the thesis and show possible significant gains. The error in tabulation of p, Figure VII, needs to be corrected. α should be listed on the Figures.

(4) The computer programs, taken in lieu of practical experiments, represent a substantial amount of detailed work. If experiments should be reported in enough detail to be critically examined and, if necessary, repeated by others, computer programs likewise need adequate documentation. The write-up for each appended program is inadequate for anyone wishing to go further on the same lines. User instructions, format requirements, statement of limitations, etc., are needed.

These criticisms, mainly of presentation, do not negate the obvious merit of the essential thesis.

by R. E. Marks

The thesis describes a good theoretical study of analysis and optimal design of pin-jointed trusses loaded into the plastic range. There are many items, however, that can be criticized; these are listed below.

1. The literature review covers a wide field. This is excellent, but in the thesis there is often an inadequate discussion of significant contributions. For example, on p.17 the conclusion of Chan is left unstated. This conclusion would be relevant to the thesis.

Other criticisms of the literature review are that some of the terms, phrases and sentences are unclear in their meaning.

(a) p.9, a clear distinction of the various energy theorems should be made.

(b) p.10, the second paragraph is vague and does not add to the development of subsection 2.2. This section requires further discussion to make it coherent.

(c) p.18, what is the meaning of the sentence starting, "Hrennikoff (1965) ..."?

2. The use of dimensionless quantities does not help the theoretical discussion. This examiner found the dimensionless quantities to be a considerable disadvantage. The use of dimensionless quantities may be good computational technique, but they should remain in the computer programs. The author should remember that he is communicating ideas; he should do this as clearly as possible. The use of dimensionless quantities, and the consequent new set of symbols, did not help in the appreciation of the theoretical work.

3. The formulae under (3.16) should be more general. The author has considered cases with internal redundants only.

4. The examples on p.28 are poorly defined. Dimensions, or even relative dimensions, are not given. More complete information on the results of the examples should have been given. The allowable stresses in each case have been omitted.

These comments apply to the other sections as well. Some discussion of results is given on p.50 under Conclusions, where discussion is not wanted.

5. In several places colloquial english expression is used, e.g., the use of doesn't and the fourth paragraph on p.30. This should be avoided in a formal report.

The second sentence on p.41 should start with "If".

6. On p.31, the word "reassuring" needs to be qualified by positive statements; it should not be qualified by chapter references only. The author's point has not been made sufficiently clear.

7. In the fourth last line on p.33 some reasons should be given why it was considered that the "first method ...... below".

8. With reference to the theory of Chapter lV, how are deflections defined so that a comparison can be made in order to find the largest deflection from the set of alternative yield patterns? Is the vector sum of deflections at a node used? Deflection should be more closely defined for a general framework analysis.

On p.39 the comments on Truss B are in conflict with those on p.39c.

9. On p.43 equation (5.7) is not written correctly. The author has a (1x2) matrix multiplied by a (cx1) matrix.

10. On p. 44 in the second paragraph it should be stated why the "efficient" design is not necessarily a m.w.d.

11. Section 5.2 refers to a single loading case. The author should have referred to the work of Sved and Drymael (reference below) who showed that for a single loading case the minimum weight frame is statically determinate. The use of linear programming for this problem is unnecessary.

12. Regarding the procedure devised for the plastic design, it seems that the author has been over-enthusiastic in applying the linear programming technique. The size of problem that can be handled conveniently by this approach is small.

If the author had minimized the weight of the truss, when all members were at the yield stress level, by varying the redundant forces, he would have been able to devise a more efficient algorithm. This approach would have required less storage than the procedure used.

The author will note that this is not the same procedure as his "efficient" design procedure (which is decidedly inefficient as comparison of the results in Fig. XV1 shows). The examples of the m.w. d's. in Figs. XV and XV1 show that each member is at the yield stress level for at least one of the applied loads.

The author should have discussed the results more thoroughly and commented on this observation. Unfortunately, also, the examples are poorly chosen; the two loading systems fully stress all members in the first examples on p.48, and almost all members in the second example.

Perhaps the main criticism is that the author did not show evidence of consideration of alternative methods of analysis of the problems. The author needs to be reminded that alternative techniques are sometimes available and may be more efficient than the first proposal considered.

13. On p.50 the second paragraph is ambiguous. The area ratios are the non-dimensionalized form of the areas: the two sentences are therefore in conflict.

14. It would have been desirable to solve more extensive problems than the trivial cases presented by the author. Although maximum sizes of frames are indicated in the programs listed in the Appendices, only by successfully running such larger frames can a guarantee by obtained that the limit is valid.

Computation times for the various examples would have been useful for future workers who might be interested in producing more efficient programs than those of the author.

Reference:

Drymael, J. "The Design of Trusses and Its Influence on Weight and Stiffness."
J. Roy. Aero. Soc., Vol. 46, 1942, pp. 297-308.