An Interactive Sizing System
for Reinforced Concrete Buildings

by

Andrew Main, B.Sc., M.Sc.

A thesis submitted to the University of Leicester
for the degree of Doctor of Philosophy

September 1973
STATEMENT

The accompanying thesis submitted for the degree of Doctor of Philosophy entitled "An Interactive Sizing System for Reinforced Concrete Buildings" is based on work conducted by the author in the Department of Engineering of the University of Leicester mainly during the period between October 1969 and August 1973.

All the work recorded in this thesis is original unless otherwise acknowledged in the text or by references. None of the work has been submitted for another degree in this or any other university.

Signed Andrew Main

Date 28 August 1973
I wish to record my great gratitude and indebtedness to Professor F. A. Leckie. His help and guidance given at every stage in the work was invaluable.

Thanks are due to Professor G. D. S. MacLellan for his original foundation of the Computer Aided Design Group of which I was a member throughout this work. The very free interchange of ideas within the Group is unique in my experience and thanks are due to all members of the Group for practical criticisms and suggestions.

My thanks are due to Mrs. Helen Sheppard for typing the thesis.

Finally, I acknowledge most gratefully the continued help and assistance given by my parents. Their part in this work has been more valuable than there is space to record.
An Interactive Sizing System for Reinforced Concrete Buildings

Summary

Computer programs for structural design are few and, largely, unpopular compared with those for structural analysis. In the interaction between user and program, there is a balance between the results produced and the time involved in using the program. The failure to provide a good balance (or worthwhile interaction) is a significant defect of most design programs. Further, non-graphical output predominates, but is often unsuited to convey results. Finally, earlier design decisions generally affect the final structure more than later decisions, but design programs do not differentiate between these stages.

A system has been implemented which avoids these pitfalls. It allows the user to find the initial sizes of members (beams, columns, slabs) in a reinforced concrete building.

The detailed design of the computer system is discussed. The structural design methods are based on the lower bound theorem. Design algorithms which test the adequacy of members are shown with their theory.

Examples of the use of the system are presented. The sizing of a set of slabs, supporting beams and columns is shown. The sizes given by the system are more quickly found and less likely to be in error than those of hand-based methods, and costs appear to be less.
## CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>ORIGIN OF SYSTEM</td>
<td>1</td>
</tr>
<tr>
<td>II</td>
<td>DESIGN OF SYSTEM</td>
<td>18</td>
</tr>
<tr>
<td>III</td>
<td>STRUCTURE DESIGN PHILOSOPHY</td>
<td>38</td>
</tr>
<tr>
<td>IV</td>
<td>USE OF SYSTEM</td>
<td>55</td>
</tr>
<tr>
<td>V</td>
<td>CRITICISM OF SYSTEM</td>
<td>83</td>
</tr>
</tbody>
</table>

REFERENCES 91

APPENDIX  FLOW CHART OF THE CROSS-SECTION DESIGN ALGORITHM 93
CHAPTER I

ORIGIN OF SYSTEM

1.1. USE OF COMPUTERS IN CIVIL ENGINEERING DESIGN

1.1.1. Introduction
1.1.2. Analysis - development
1.1.3. Design - development

1.2. DESIGN PROCESSES AND INTERACTION

1.2.1. The need for interaction
1.2.2. Qualities of interaction

1.3. REAL-TIME INTERACTION

1.3.1. Historical approaches to interaction
1.3.2. Duplication of the design processes
1.3.3. The need for special programs

1.4. SIZING BUILDINGS - BESS

(Building Engineer's Sizing System)

1.4.1. Present design procedure
1.4.2. Reasons for choosing buildings
1.4.3. What to expect from a sizing program
1.1. USE OF COMPUTERS IN CIVIL ENGINEERING

1.1.1. Introduction

The ambition of removing burdensome, tedious arithmetic from civil engineers by passing it to computers is nearly twenty years old, but the early optimism has still not borne fruit in all areas. Though great strides have been made in the analysis of structures, for which programs abound, design is still very much a man-based procedure. The progress of analytical methods is such that most civil engineers regard analysis as a financial rather than a technical problem. By contrast computer programs specifically intended for design are very few. Indeed the assumption that speedier analysis will diminish or even dissolve design problems underlies most of the early engineering programs and is, in large measure, the reason for the lack of success of such design programs as have been written. One cannot design without analysis, however, and it is essential to be aware of analytical programs and their history before examining the field of design.

1.1.2. Analysis - development

Analysis has progressed ahead of design for two major reasons. It is a precise science which obeys logical laws, whereas design is subject to human judgement. Design cannot overtake analysis just as the cart cannot overtake the horse. However, the earliest programs were kept as a last resort by most designers because primitive input and output media and lack of 'data generation' facilities within the programs made them difficult to use. Therefore the advances have not been
limited to technical considerations, since computer programs have progressed to make the new techniques easier to use and more convenient to encounter.

Since then considerable effort has gone into program development to make programs less inhibiting. One example is the use of new hardware to add flexibility to the programs, such as the use of disk storage to pass data from one to another phase of the calculation. The programming effort has not quite matched the advances in techniques and much remains to be done on the 'convenience' side. Despite the gap, however, analysis of structures actually built today is commonly regarded as a 'solved' problem except for structures which are now so uncommon they are 'one-off's' (e.g. long span suspension bridges), for which a one-off program is written, if necessary.

With analysis handled so well by computer hardware a natural ambition is to produce and use design programs. In fact there have already been attempts to do so. These attempts should be studied before attempting the design of a new system.

1.1.3. **Design - development**

It is very rare to find an engineering computer bureau without a program for analysing plane frames. It is just as rare to find one with a program specifically written for design. Instead of programs, the past has seen the growth of 'systems'. These systems are designed to remove the need for specialised computing knowledge and to allow easy development of design programs. The help comes in the form of powerful programming tools and easy means of controlling them from within the program. SPAN and ICES are two such systems. SPAN, the grandfather of
GENESIS, was produced for the private use of consulting engineers whereas ICES was written at MIT and was intended for widespread use.

Together with most early systems, ICES and SPAN were based on what might be called the 'alternative design' theory. The theory is that faster, cheaper, more convenient analysis results in more cases being analysed in a given time (and for a given sum). With more information at his disposal the engineer makes his choice more easily. For reasons which will now be illustrated, this has not proved entirely correct.

The systems all offer three major facilities: some command-oriented language, a data-base, and selective output, are common to all. There are minor bonuses with each individual system but they are of no great concern. The command-oriented language was introduced to make input more intelligible to engineers. Backing store has long been used to pass data between programs of a suite and the systems all have a data-base to allow different programs (more often called 'sub-systems') to use the same data. This is useful when the gross data, say for a bridge, contains information suitable for two or more programs, such as column design and foundation analysis. Selective output was introduced to pacify engineers who objected to wading through oceans of output.

In spite of the high hopes of the designers, the systems still remain little used because they are too cumbersome. SPAN, which is one of the smallest, provides a good illustration of this. Written to avoid the major pitfall of ICES, which is the 'system overhead cost' particularly in the input phase, SPAN validates all the data for a job before execution. This reduces the number of costly runs in which a fault late in the data aborts the whole program. The method, however, precludes SPAN from being used via an 'interactive' medium such as a Teletype. It is a 'batch' program and therefore has slow turnaround. Although at a
particular stage an engineer, designing by hand, might take many very small steps to determine what to do next, he would naturally be reluctant to wait for SPAN to do the same because of the greater time involved.

All the systems either hinder engineers, or cost them so much that they prefer to perform some operations (though not all) by hand. This is not the only reason for the lack of use of such systems but it is important and the nub of the matter is in their failure to mirror the engineer's own methods of design. They offer ingeniously controlled brute-force, but the engineer does not always want brute-force no matter how skilfully it may be directed. A counterbalanced sledge-hammer will drive six-inch nails with ease but is still painful to use on panel-pins.

Thus the systems in use for some years do not correspond with the designer's own man-based methods. The next section discusses these methods and what the computer needs to offer before it can properly compete.
1.2. DESIGN PROCESSES AND INTERACTION

1.2.1. The need for interaction

Structural design is not accomplished by following some well-defined path. Each engineer cuts his own way through to the final design using non-scientific (or at least unquantifiable) faculties such as invention, intuition, experience and judgement as well as technical knowledge. It is well known that neither the final designs nor the methods used by different engineers will necessarily be the same, yet it is also true that neither result nor method is any less valid.

Design involves exploration in its earliest stages using iterative trial-and-error procedures, and becomes more straightforward, as the design progresses. Initially, many attempts are made to define the critical points of the structure, those points which affect the remainder more than most and whose final details often largely shape the remainder of the structure. Some of these attempts at definition embrace large scale operations such as frame analysis, but many encompass lesser operations whose purpose is just as important.

In these processes time becomes a major criterion: the engineer wants the results of some action to be proportionate to the time spent on it.

The same applies to his use of computers. He seeks them when they offer a good balance between time and results. However computers involve additional time burdens, that of preparation and that of waiting for the job to be done (if submitted in a batch queue). It is these two times which distort the time-result relationship and cause the engineer to reserve for the computer those jobs whose size merits a long wait.
That is to say he does the smaller jobs by hand, leaving only the large ones for computers. Even so, the batch queue is not the only deterrent, for to use the computer via, say, an electric typewriter, a choice has to be made between two unattractive alternatives. He may on the one hand use a 'one-off' type of program which will involve entering a lot of data, or on the other hand he can opt for a system which retains the data from run to run but which will prove expensive on input (ICES) or be unsuited to interaction (SPAN) or be less expensive and more suited to interaction - but only slightly so (GENESYS).

What has really been discussed here is interaction in its broadest sense. When using a computer one is always interacting with it. It performs some action, in a certain time, and one responds after a period of thought; this cycle is repeated until a decision is made.

In other words the time taken by the machine to respond is generally too long for it to be given small tasks. This is a major cause for computers being on the periphery rather than in the main stream of engineering design. Not until the response times are reduced can there be any expectation that the situation will change. The limitations of the batch queue will in many situations prevent satisfactory interaction and for these situations interactive programs are necessary.

1.2.2. Qualities of interaction

For programs to be an integral part of design they must offer the full range of response times. There is no point in making every program so fast it responds immediately. If the engineer's thinking time between runs is two months, then clearly it does not matter to him if the program needs an hour or a day (or even a week). As already stated, he
expects the time to be in proportion to the work done and as his own thinking time is likely to be in proportion to the work done (so that he may fully understand the results) there is little to be gained from accelerating response beyond a certain point. The necessary and sufficient condition is that he should not feel he has waited too long. If that condition is fulfilled, the engineer is satisfied (with the proviso that the program gives him what he wants).

Formerly, computer operating systems were not geared for interactive work and response times varied greatly depending on the amount of work going through the machine. A limited quantity of interactive users were satisfactorily served, but the limits were lower in practice than in theory. For instance SCOPE (CDC 6000) permits 63 users but response degrades badly at 15 on a CDC 6500. Nowadays, however, operating systems have been written that cope well. KRONOS (CDC 6000) has an upper limit of 512 and the author has used a 6400 with 128 successfully attached and used. (A 6500 has two central processors, a 6400 has only one).

Interaction, then, is not only necessary but is also achievable with modern equipment. It must be designed to satisfy the engineer at all points so that his 'time-balance' is satisfactory, but at the same time the programmer must not sacrifice all else just to reduce response times. There is no point, for example, in speeding interaction by distorting engineering logic.

The interactive program must offer engineering satisfaction. It must communicate effectively with the engineer and must respond adequately. The first of these is achieved by allowing the engineer:

(a) to choose the appropriate method of analysis

(b) to take all decisions, particularly those ill-suited to the computer
(c) to leave decisions to the computer. (He should not be forced to take every single decision, often a default value will be good enough.)

The second is the more complex. Interfaces between men and machines are generally heavily reliant on the written word, either typed or displayed on a Visual Display Unit (VDU). This is not, in itself, adequate. For structures, the maxim that a picture is worth a thousand words is undoubtedly true. (Design offices do not have the synonym drawing offices for nothing.)

A 'hard-copy' on paper of some information is still necessary but the Graphical Display Unit for example has immense value for interactive structural analysis programs. The graphical display is specifically mentioned because it satisfies the response time requirements (the third provision) better than plotters, microfilm devices and the like.

A combination of graphical display, alphanumeric keyboard and line-printer constitutes a likely set of ingredients for success. It would allow pictures to be transmitted and numbers to be entered as well as the recording of quantities of data. Some plotting device would clearly be a useful addition for pictorial records.

It is surprising that systems (such as ICES, GENESYS, et al) are ill-suited to use such equipment when their designers lay such stress both on graphical output and on teletype usage. However, batch processing heavily predominates over interaction commercially. In consequence, systems are designed around batch operation. Only superficial provisions are made for interaction. For example, output may optionally be restricted to the width of a teletype. This is not enough. Programs (and subsystems) deal with real situations through models. The unity of program and model results in the mutual shaping of either one by the other.
It is therefore highly improbable that models designed for batch operation will suit interaction, since the requirements of the two modes of processing are so different. For example, since most past systems are based on a command-oriented language, the interactive user is confined to the alphanumeric keyboard as the sole, active interface to the computer. Any displayed information is degraded to the level of an 'extra' because the user is constantly distracted from it. For good use of the display he should rather be permitted to concentrate.

To summarise, the program that is designed to help the engineer to the full at every stage of his design must afford him interactive responses geared to his conception of the work involved. It must further allow him to use a graphical display screen if his requirements so dictate. A program which uses the display must be specifically designed to make best use of it.
1.3. REAL-TIME INTERACTION

1.3.1. Historical approaches to interaction

In general, real-time interactive programs fall into three categories:

1) "Alternative design"
2) "Design variable"
3) "Tailor-made"

As stated, most of the early systems are of the "alternative design" type. They allow the engineer to try out many designs very quickly by providing fast analysis and easy-to-use input. They fall down in the realm of real-time-interaction because of slow reaction times and deluges of (non-graphical) output.

The "design variable" is the emble of the SPAN-CLEAT-GENESYS dynasty. It is an extra feature which allows the user to enter any or all of his data as a named quantity. Before running he must put values to the various named quantities. He may re-run, as often as he likes, with a different set of values. Unfortunately this idea is too simple. It is difficult to optimise functions of more than two independent variables (if one accepts design as a kind of optimisation), and if the problem depends upon two (or one) variables it can usually be solved by hand. (The author has worked for some time alongside users of this dynasty and has never found a single user of the design variables!)

"Tailor-made" systems have predominantly been intended for batch use. An example is the "KIF" system for detail design of a (restricted) type of building. It has been found to be useful within the construction industry and sparked off one or two other programs (e.g. GENESYS/BUILD).
Such systems are certainly "possibles" in the field of design but so far have not been written for real-time interactive programming and a batch intended system will never be anything but cumbersome and inconvenient if used interactively.

1.3.2. Duplication of the design process

In design the engineer progresses to his final structure along a path which can be simplified as follows:

a) Conceptual design:
   At this stage his ideas range widely over the possible different ways of solving his problem e.g. Steel/Concrete; Flat slabs/Ribbed slabs; In situ/Precast.

b) Sizing:
   Having decided to a large extent how to solve the problem he determines the sizes of members needed to support such a structure.

c) Detail design:
   As its name implies this stage embodies the exact calculation of details of the structure.

These areas are not well defined. Stage (a) involves quite a lot of stage (b) and even some stage (c); stage (b) involves some stage (c), (though (a) should be complete); stage (c) is on its own when (b) is complete. The problems that arise initially (in (a)) have a certain character. In buildings they concern matters such as ceiling clearances, heating ducts, architect's needs of inter-column spacing. For bridges they touch on support positions (and restraints thereon), head-
room and the like. The character of these restraints is topological, they concern gross, positional aspects and positional relationships. It is the engineer's job to resolve their conflicts.

As the design progresses the problems become more geometrically precise. They involve reinforcing bar arrangements, weld patterns and such. It is worth noting that if the early decisions are wrong, the engineer may reach an impassable problem and have to retreat, always a costly procedure.

As stated, this design process must be duplicated in programs as well as possible for engineers to use them readily. Despite this, it is not necessarily essential for the computer to use exactly the same methods as the engineer in every situation. For example, one may consider the problem of choosing column sizes in a building. It is an enormous task in a complex beam layout to calculate by hand the exact load which is carried by any one column. In the drawing office, therefore, an approximation is made, usually based on the area of flooring which the engineer feels the column carries. (Provided the sum of loads he expects each column to carry is greater than or equal to the weight of the whole floor and its live loadings, he feels reasonably confident. If he does not feel so, he does a grid-slab or plane frame analysis). By contrast, it is not difficult to write a program which computes such information precisely and quickly. Provided the speed is consistent with expectation, such a program will be most satisfactory even though it does what the engineer would never dream of attempting.

1.3.3. The need for special programs

The hardware specified in Section 1.2.2. applies restraints to the
programs which are written for design purposes. For refreshed graphics a picture must be held in digital form in the machine so that it can be scanned several times per second (e.g. thirty, forty). Moreover, for quick picture modification all the data must be available with the shortest possible delay; everything must be done to ensure the maintenance of speed throughout such stages. In addition, the means of communicating choices to the computer have to be selected at each stage. Choosing one path rather than another may be managed by depressing a "function key", indicating with the "lightpen" a certain item which is displayed on the screen, or typing on the alphanumeric keyboard. In some cases two of these possibilities may be open at once, in others - all three. All these needs must be catered for within the program in a 'transparent' way. The user must not be aware of the catering, only that he has been catered for.

The program must allow trial and error so that the designer can go backwards or forwards, to try little pieces of design 'out of context', to carry through to detail state in one area, leaving an adjacent area alone, if he so wishes.

The structure of such a program is very different from any other interactive or batch program. The data is likely to be 'paged' on and off disk. The paging may be multi-level with directories to speed searches and will certainly be structured for speed of reference so that the program can quickly locate connected items. Being a large program it will be likely to be segmented or overlaid.

These are general points, but it must be remembered that each 'reality' for which a program is written (e.g. buildings, roads, dams) has its own model which needs its own paging, its own structure of the data (reflecting interrelations), its own paths through the program and
its own way of presenting and amending pictures.

The particular problem which was chosen as the subject for study is the sizing of reinforced concrete buildings.
1.4. **SIZING BUILDINGS - BESS**

*Building Engineer's Sizing System*

1.4.1. **Present design procedure**

The broad outlines of design procedure have been illustrated, the progression from conceptual to detail design being shown in Section 1.3.1. Such procedures are not available in practice. For example one of the penalties of present construction speeds is that designers are often only a few steps ahead of constructors. In the design of buildings it is desirable to move from top to bottom, but since the foundations must be constructed first the order of design is reversed and this approach depends entirely on flawless calculation of initial sizes. Sizing is hurried, though, and must at best be crude with mistakes likely. Mistakes are expensive involving modifications and delays.

Clearly there is benefit to be gained by accelerating the speed of the initial sizing process and by removing inaccuracies.

1.4.2. **Reasons for choosing buildings**

The design of reinforced concrete buildings constitutes a large portion of design office work. Though such structures are not the most interesting they are numerous and are an obvious choice for a sizing program.

A further reason for choosing buildings is that they have many common features and therefore the program will not need to be a conglomeration of different, smaller, programs each needing to be understood by the user. Instead it can be unified and simpler to use.
1.4.3. **What to expect from a sizing program**

Assuming that the program is technically competent the following is a summary of the important features of any program intended for sizing reinforced concrete buildings.

a) The engineer must be in sole command but must not be forced to take every decision at every stage of the program's execution.

b) The engineer must not feel he has waited too long for machine response at any stage.

c) A graphical display screen should be included in the hardware, because of the need for fast transmission of information.

d) The engineer must be given appropriate analysis methods which keep him as well informed as hand-based methods.

e) The interaction interface should be transparent.

f) As many anomalies of building structure should be catered for as possible.

g) The system must allow quick examination of alternatives.
CHAPTER II

DESIGN OF SYSTEM

2.1. INTRODUCTION

2.1.1. Limitations on building types
2.1.2. Hardware and software
2.1.3. Data of buildings
2.1.4. Required interaction times
2.1.5. Summary

2.2. MODELLING BUILDINGS

2.2.1. Data structure
2.2.2. Algorithms
2.2.3. Need for paging

2.3. PAGED MODEL

2.3.1. Paged data structure
2.3.2. Algorithms for paged model
2.1. INTRODUCTION

2.1.1. Limitations on building types

Before describing the design of the system it is essential to study the features of the design process and computing facilities. The calculations involved in sizing members of a building follow a simple pattern:

a) Determine the necessary slab depths for the various sets of slabs on the floor under consideration, from which weights can be found.

b) Work out the distribution of slab weight and loading onto supporting members (typically beams) and determine their various necessary dimensions (widths and depths).

c) Work out the distribution of floor loads between the columns and determine the necessary column area.

This procedure is conducted from the uppermost floor to the lowest. At present many different types of analysis are used. For convenience most are rather simple, but in complex buildings a full 3-D framework analysis may be necessary. When buildings are structurally complex, sophisticated analysis is necessary. The great majority of buildings are simple, however, since complex buildings are expensive to construct as well as difficult to design. Simple buildings can be quite complicated but they satisfy certain rules. For example, the column centrelines do not "wander" from floor to floor and the beams tend to meet orthogonally. The design of the system is limited to buildings of this type.
2.1.2. Hardware and Software

In order to provide the interaction feature described in Chapter I the data must be immediately accessible to allow information to be presented to the user quickly. There are, of course, many possible solutions to this problem.

The choice is governed by the available hardware, the available software, the chosen mode of interaction, the size and character of the data, and finally the speed of interaction. These are not independent of each other. The software depends upon the hardware, while the interaction mode depends upon the hardware, software and the data. At the time of design of the system the University/Computer was the only available hardware. Its configuration was as follows:

a) 64K of 24 bit words
b) 4 Magnetic disk handlers
c) 8 Magnetic tape handlers
d) Graphical display with lightpen, function keyboard and teletype
e) Digital plotter
f) Line printer (slow)
g) Card reader (slow)
h) Paper Tape reader and punch

The software available for use on the machine comprises FORTRAN IV, ALGOL, NEAT (assembly language) and software at the FORTRAN and ALGOL levels (subroutines and procedures) and the NEAT level, to allow the display to be operated.

NEAT was disqualified because it was very difficult to use in a batch environment. ALGOL and FORTRAN were compared at the start of this project and ALGOL had the following advantages:
a) Transfer rates (of data) to disk are 25 times quicker in
ALGOL than in FORTRAN.
b) Segmentation of code was not working in FORTRAN though it
was in ALGOL.
c) ALGOL was more reliable, on the 4130, than FORTRAN.
d) The teletype was not usable with the display with FORTRAN.
e) The ability to allocate and return arrays in ALGOL is only
mirrored in FORTRAN by a 'free-storage' scheme.

2.1.3. Data of Buildings

A preliminary study of buildings produced statistics concern­
ing the size of data. For a 1000 column building there are nearly
300,000 items of data. (A column is defined as running from one floor
to the next) A building of such size is in the small to medium range
of buildings. This data describing buildings can be broken into six
categories:

a) External geometry of members.
b) External loads on members.
c) Internal forces (i.e. Bending Moment, shear, thrust).
d) Material properties.
e) Reinforcement details.
f) Boundary conditions (fixity).

Additionally, there is data which describes the connections be­
tween members. This data is 'associative'. It describes the associa­
tions of one member with its neighbours. 'Associations have an impor­
tant effect on the solution of the problem.
2.1.4. Required interaction times

The speed of interaction is important. In purely subjective terms, times can be classified in the following way:

<table>
<thead>
<tr>
<th>Time (Seconds)</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; \frac{1}{10}</td>
<td>Instantaneous</td>
</tr>
<tr>
<td>&lt; 1</td>
<td>Hiccough</td>
</tr>
<tr>
<td>&lt; 2</td>
<td>Pause</td>
</tr>
<tr>
<td>2 - 5</td>
<td>Wait</td>
</tr>
<tr>
<td>&gt; 5</td>
<td>Long wait</td>
</tr>
</tbody>
</table>

It has been found that users will tolerate delays up to two seconds ('pause') before their interest turns to frustration. During purely graphical operation it becomes very tedious to have to scan a screen to see if a change has occurred. The teletype can be used to emit a noise, therefore, after 'waits', to reduce tedium. Despite this the aim should be to keep all delays down to two seconds.

2.1.5. Summary

All of the factors introduced here clearly bear on the solution of the problem and they have to be related. The remainder of this chapter illustrates the reasons behind the solution chosen, and does so in two parts. First a solution is shown suitable for 'in-core' operation. Following that the need for and consequences of the use of backing store are discussed.
2.2. MODELLING BUILDINGS

2.2.1. Data Structure

The treatment of buildings in a computer requires the ability to manipulate the data which describes the building. Such manipulation requires a formal digital model. D.T. Ross defines a digital model as consisting of three parts, the data, the data structure and the algorithm.

In order to explain the model, it is necessary to illustrate one method of comprehending data structures. This method has been used for some years and is based upon 'beads' and 'pointers'.

Computers' core store consists of a contiguous set of 'words', each having its own 'address'. Addresses are integer numbers which range from zero upwards. The top address is, typically, 32767, 131072 (or some other number).

A 'bead' is defined as a group, of any length, of such words. The 'pointer' to any bead is defined as the address of that word (in the bead) which has the lowest address. The power of this concept lies in the fact that beads may store both data and pointers. For example, if one bead holds the data concerning a column, as in Figure II.1, then ten extra words may be added to the bead to store pointers to those other beads which represent the members to which the column is attached. In this case the ten members comprise two columns and eight beams as shown in Figure II.2. The new data structure is shown in Figure II.3. By convention, if a member is missing the corresponding pointer might be given the value zero. This method gives a very flexible method of modelling the topology of buildings.

There follow a few examples of beads and the use of pointers.
<table>
<thead>
<tr>
<th>LENGTH</th>
<th>1st BEAM</th>
<th>2nd BEAM</th>
<th>3rd BEAM</th>
<th>4th BEAM</th>
<th>5th BEAM</th>
<th>6th BEAM</th>
<th>7th BEAM</th>
<th>8th BEAM</th>
<th>1st COLUMN</th>
<th>2nd COLUMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BREADTH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THICKNESS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATERIAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of bars of steel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of steel stirrups</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOAD</td>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
<td>4th</td>
<td>5th</td>
<td>6th</td>
<td>7th</td>
<td>8th</td>
<td>1st</td>
<td>2nd</td>
</tr>
</tbody>
</table>

**FIGURE II.3**
Figure II.4 is a representation of the core store of a computer (with 32K words of store). Four beads are shown.

- a) A bead of 1 word whose (address) pointer is 10.
- b) A bead of 1500 words whose (address) pointer is 10000.
- c) A bead of 1000 words whose (address) pointer is 16000.
- d) A bead of 6767 words whose (address) pointer is 26000.

There are other definitions of pointers, which are not based upon the address of the first word. If the bead lies within, say, an ALGOL array, then its pointer is the array index of that word which has the lowest array index. The pointer is the index of the first word.

Values can be stored in the beads and retrieved from them. This is done using an 'offset' from the top of the bead. If a required value is known to reside in bead (c) at its 10th location then the value may be retrieved by examining the (16000 + 10th) location. The merit of this method of reference, in computer terms, is that the value of the pointer need not be known, provided its value can be found. For instance, if variable I is set to the value of the pointer (16000), the value previously
required is to be found in location \((I + 10)\). If many beads of the same type (i.e. representing objects with similar characteristics) are in existence, their layout can be standardized: their first characteristic stored in position 0 of the bead, the second in position 1, and so on. The consequence of this is that the characteristics can be stored and retrieved very simply. If variable \(\text{CHAR}\ 11\) is set to the value of 10, then the tenth value in bead (c) is found in location

\[(I + \text{CHAR}\ 11)\]

If \(J\) is set to the value 10000, then the 10th value of bead (b) is found in

\[(J + \text{CHAR}\ 11)\]

Moreover, these values need not just be characteristics (such as \((X, Y, Z)\) coordinates), they can be pointers, so if the number 16000 is placed in

\[(J + \text{CHAR}\ 11)\]

the tenth position of bead (b) points to bead (c). The examples just given should aid understanding of the techniques used to provide a model.

Beads and pointers are the basis of the data structure used for the building sizing program. Since beads can be of any length, a choice has to be made on how much data should be stored in one bead and what it should describe. The obvious choice is to use a different bead for each individual column, beam and slab. The data to be stored is not simply the members' interconnections and external dimensions, since details of reinforcement must also be included. Reinforcement is rarely confined to one member, it is often common to two or three. An example is the steel passing over a beam into two slabs. The data describing the
Steel need only be stored once, but if it is assigned by convention to either slab, a problem arises if that slab does not exist. Assigning the steel to the beam, which must always exist, solves this problem. There is a corresponding problem concerning the junction of beams at column heads. For all such problems the strategy was adopted that data concerning such steel should be stored with the member of higher order. The order of members is:

- Highest: Column
- 2: Node
- 3: Beam
- Lowest: Slab

An extra member had to be introduced to join beams to each other and to columns. It is called a NODE and represents a 'joint' in structural terminology. It can be seen that the items in the table are structurally dependent on those higher in the table, this is echoed in the data structure. Nodes are linked across floors by beams and slabs and linked from floor to floor by columns. Diagrammatically the situation is as shown in Figure II.5.
The arrows in Figure II.5. are shown double-headed to avoid having twice the number of single-headed arrows. The arrows represent pointers in the data structure. In the diagram node N1 is the central part of both the data structure and the actual structure. It points to the upper column C2 and the lower column C1. They point back to N1 and also to the upper and lower nodes N2 and N3. Node N1 points to its four beams B1, B2, B3 and B4, which all point back to it. Each beam is shown pointing to one slab, but in fact would point to two. The extra slab has been omitted for clarity.

Figure II.6. shows the data structure more clearly, leaving out the actual structural components. However this structure is unnecessarily deficient in its descriptive power. It only allows slabs the support of a column at each corner, whereas slabs frequently have several columns along one edge. A typical arrangement is shown in Figure II.7. The data structure, though not allowing for this, can be adjusted to do so. The problem arises in slab definition, so slabs are re-defined. A slab is no longer defined by a maximum of four edge beams, but instead is defined by a maximum of four corner nodes. Now any number of beams may run along a slab edge. The data structure needs extra pointers, so that slabs may point at the corner nodes. The use of the pointers must change because beams may now point to the same slab as their neighbour. The beam-to-slab pointers stay the same but their use changes. Algorithms use the data structure and they are discussed in the next section.

2.2 Algorithms

Algorithms are the logical processes used by programs to access data within a data structure. They are an essential part of any model.
Legend

\[ x = \text{COLUMN} \]

\[ - = \text{BEAM} \]

Plan view of slab

FIGURE II.7
This section discusses the algorithms needed for the data structure which has been outlined and indicates improvements in the data structure that would save program space and time.

The data structure shown in Section 2.2.1 is good on a small scale, but poor for large amounts of data. Provided the pointer to a node is known, it is possible to determine how many beams, columns and slabs exist around it. It is very difficult to find the pointer initially. The problem arises from the flexibility of the data structure. It can describe a building that is 'U'-shaped in plan, as easily as a regular box-shaped building. The algorithm must be capable of performing the following function: given the pointer to any node it must 'find' some other node which may be specified, for instance, by its spatial coordinates. Clearly the algorithm to do so would have to be capable of finding its way round a three-dimensional maze. A simple solution to this problem is to modify the data structure so that simpler algorithms may be used.

Two solutions come from classical data structure theory. In the first each node is placed on a 'list'. An extra pointer is attached to each node. The new pointer points to any other node, provided that the other node is not already being pointed at in this way. In the second solution a directory is used. This contains a few items of data, e.g. its spatial coordinates, which allow any node to be uniquely identified, together with a pointer to the node itself. These solutions both leave the rest of the data structure untouched so that the flexibility is unimpaired. The algorithm to find any one node is much simpler using either of these solutions. Consequently the program will be smaller and quicker.
2.2.3. Need for Paging

The time taken to search a list for a given node depends upon how far along the list the node occurs. However, if the processing of each node by the search algorithm takes, say, 100 \( \mu \) seconds and there are 6000 nodes in the building the average search time would be 0.3 seconds. These times are based on the assumption that all the data is in core. It will be recalled that in Section 2.1.3. the size of data was stated and so much data cannot all reside in core. Backing store must therefore be used. The remainder of this chapter discusses the effect of the use of backing store on the model.
2.3. **PAGED MODEL**

2.3.1. **Paged Data Structure**

The use of backing store affects the data structure and algorithms in a marked, but not drastic, manner. The two media for storing the data structure are magnetic disks and magnetic tapes. Magnetic tape works much like a tape recorder, but records binary bits of words and allows these patterns to be 'read' back into core store. Magnetic disk is more like a record having read/write heads which can 'hop' from one track to another. It records bits, like tape, and they can be read back. Disks hold less information than tapes but allow much quicker access to it.

Data is transferred between core and backing store in the form of 'pages'. The principle of paging is to break data into pages which can be examined individually - as the pages of a book. (Though one cannot see the entire book, several pages may be viewed together.)

The choice of a paging strategy depends on a number of interdependent factors. These are:

a) The size of pages.
b) The data to be stored in pages.
c) The pointers to be stored in pages.
d) The size of available core.
e) Required access times.
f) The data structure to be paged.

The problems associated with each of these factors are, respectively, as follows:

a) If pages are small, much searching of backing store is needed.
Searches are slow. If pages are large, they may have to be transferred to and from backing store frequently, to make room for each other. Transfers are also slow.

b) If data is interrelated a problem arises at the page boundaries, where there must be a split in the data. Correct boundaries are those which minimise inter-page references.

c) If a pointer refers to an item within a page (intra-page) its form is simple. If a pointer refers to an item in another page (inter-page) it must be formed from two parts. One identifies the page, the other identifies the item within the other page.

d) If space for programs is decreased, more is available for data and vice versa.

e) If the average access time for data is crucial a different scheme may be needed than if, say, the time for the worst case is crucial.

f) If the data structure is of the 'list' type a different scheme would be needed than if a directory is involved.

Relating these factors is clearly difficult and there are many solutions. Since no formal paging scheme existed at the University when the system was being implemented, one had to be created. It was decided that pages would all be of the same size, rather than of varied sizes. This simplified the writing of the paging scheme. For example, each bead has a fixed position in the page and therefore the use of intra-page pointers is easier. In addition the pages are placed in rows of a two-
dimensional ALGOL array. Some of the control of paging is therefore handled by the ALGOL compiler.

An important corollary of constant page size is that the data in each page will describe a basic unit, or "building brick" from which entire buildings may be modelled. Three basic units are shown in Figure II.8.

Isometric projection

![Isometric projection diagram]

Figure II.8.

Multiples of these same units can be used to produce larger components as shown in Figure II.9.
Both the B and C units share information with other units. The shared information can be stored in separate pages, otherwise the neighbouring units must be in store simultaneously. Clearly until the B and C series become large multiples of the basic units a great deal of space is wasted in duplication of data. The A series can be kept small and will allow many neighbouring units to be core resident simultaneously, something the B and C series can only do if kept small and therefore inefficient. For this reason A was chosen as the basic unit and consists of a single column, two beams, nominally at right angles along two edges of a slab.

The A unit is flexible and efficient with regard to space and computing time. The same data structure can be used as that outlined in Section 2.2.1. with the additional features needed for paging. The changes that have to be made to the data structure to adapt it to paging
are quite simple.

Six inter-page pointers must be added to each node. Additionally, the pointers from slabs to nodes and from beams to slabs have to be inter-page instead of intra-page. Six extra pointers are required because any unit has a maximum of six neighbouring units. They lie in the directions North, South, East, West, Up and Down; these are the names given to the pointers. (As in the unpaged model this limits the number of beams meeting at a point to four, though the maximum could be overcome by using dummy units of the same coordinates as the real unit.) The alteration of the other pointers from intra-page to inter-page occurs because in the general case the member being pointed to will not be in the same page as the one pointing. Sometimes an inter-page pointer will refer to data within its own page. The overhead involved is more than compensated for because all accesses of data can be performed by one routine.

Figure II.10 is a diagram of a page in the final structure. Inter-page pointers are shown double-headed. Intra-page pointers are single-headed, but for clarity not all intra-page pointers are shown. The actual building members are also shown.

This, then, is an overall view of the data structure part of the model, outlining the reasons why this model was chosen. The algorithms for the paged model are discussed next.

2.3.2. Algorithms for Paged Model

One of the most important requirements of paging is that it should be quick enough for interactions of the necessary kinds. A study of the uses of the system is therefore essential.

The initial creation of the model is performed by a 'batch' type
Final Data Structure

FIGURE II.10
of program which accepts data on cards and generates the data structure on backing store. Interaction is carried out under control of a separate program. The input program sets up all units in the data structure together with their links. It further generates grid lines along the X-axis, another along the Y-axis and one for floors up the Z-axis. Points within the building may therefore be specified using this grid system. Checks are carried out which ensure that the model is realistic.

For the interaction program the various operations which are necessary are as follows:

a) Draw floor plans or sections.

b) Input member loads.

c) Print out details of members.

d) Input data direct to units (over-riding that selected by the program).

e) Modify the structure by moving members.

f) Size a plane of slabs.

g) Size a continuous beam and its supporting columns.

A further useful feature would be to be able to alter the structure by adding or removing members. The use of this would be limited but nonetheless valuable.)

There is a pattern of data access in the operations listed above. Either units are accessed singly, or units in a certain plane must be processed. Operations a, f and g clearly involve members which lie in a plane. Operation b also does, though not always. Slabs on a given floor, for example, often bear the same live load. The other operations, c, d and e operate on single members. The most common needs for the algorithms are therefore:

a) To seek individual units.
b) To search for units in a given plane
and c) To access data on a small scale.
The last need was outlined in Section 2.2.2. and it is important to examine the effects of paging upon this kind of access.

To access all information about one node, seven nodes are needed (the node itself and its six neighbours). Assuming that there is room for seven nodes to reside in core simultaneously, the worst case for accessing an entire node occurs when seven units have to be transferred to backing store before the desired seven can be brought into core. The worst case involves fourteen transfers and the average involves seven, though what the statistical mode might be is impossible to estimate. The table in Figure II.11 shows a comparison of predicted times for single units and for sets of units with their worst and average times. It only shows times for disk. Magnetic tape is quite out of the question. (For comparison, the worst case for a set on Magnetic tape is 17 minutes 58 seconds.) The times are for a building of 1000 columns.

<table>
<thead>
<tr>
<th>Average for single</th>
<th>Worst for single</th>
<th>Average for set</th>
<th>Worst for set</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.060 seconds</td>
<td>0.110 seconds</td>
<td>0.42 seconds</td>
<td>1.5 seconds</td>
</tr>
</tbody>
</table>

Figure II.11.

In tests, the estimated times have been shown to be close to actual times. These times are tolerable for interaction, the worst being less than 2 seconds.

For the random access of units within the data structure a list or directory is needed. Long lists are prohibitive for paged data struct-
ures. In Section 2.2.3, it was shown that 0.3 seconds would be needed for the average search (if all data were in core) through the data of 6000 nodes. If the data were paged this time would jump to two minutes. A directory is the solution to the problem. The searches for units are either for nodes in a plane or for one node at a specified grid reference. As grid references decode into geometric coordinates, the directory is based on the coordinates of each unit.

The directory stores the disk location and coordinates of each unit. Searches can be quickly carried out, therefore, in a fraction of the time needed to complete the actual transfer of data. Consequently, the data cannot be paged more quickly. Unfortunately, to prevent it from occupying too much core, the directory has to be paged itself. This slows the process down, but a very substantial saving of time is still achieved.

For large scale operations, such as drawing whole sections, the times are likely to be too long (of the order of 10 - 15 seconds), but an engineer will not generally wish to skip from view to view, since he will examine one view for some time before moving to the next. For smaller scale operations time should be almost instantaneous.

It should be noted that even when the program is active and is not waiting for the engineer, it is 'channel-bound'. In an interactive environment costs would therefore be expected to be low.

The exact nature of the algorithms cannot be shown for reasons of space, but in any case their general method, the times they take and the reasons for their choice are more important. The basis of the model has, therefore, been covered. The data, its structure and the matching retrieval algorithms have been discussed. The next chapter treats the engineering analysis and design processes adopted for the system.
CHAPTER III

STRUCTURE DESIGN PHILOSOPHY

3.1. THE DESIGN METHOD

3.1.1. The members to be designed
3.1.2. The choice of lower bound methods

3.2. THE LOADINGS

3.2.1. Introduction
3.2.2. Distribution of loads in slabs
3.2.3. Loads on beams and columns

3.3. THE TESTS OF MEMBERS

3.3.1. The limits of bending moment, shear and thrust
3.3.2. Tests for cross-sections in bending and shear
3.3.3. Handling cross-sections in bending and thrust

3.4. USE OF LOADING AND TEST PROCEDURES

3.4.1. Slabs
3.4.2. Beams and columns
3.1. THE DESIGN METHOD

3.1.1. The members to be designed

The model can describe buildings which are composed of beams, columns and slabs. Each of these members is itself described within the model. However, some limits had to be placed on each member's description. The limits have to be wide, to allow the descriptions to cover as much as possible, but they must not take up too much space. The descriptions are illustrated before discussion of the chosen method of design. Though the method is general, the discussion is restricted to the three members to which it is applied. Slabs are shown first, followed by beams and finally columns.

Slabs are supported on all four sides by beams. They carry loads in two directions and are correspondingly reinforced. Steel is provided to resist the sagging moments between supports and the hogging (or 'reverse') moments over the supports. Shear force is not a direct problem in slabs. It affects 'local bond stress' only. All moments in slabs are resisted by single layers of steel in tension. Slabs have a constant depth.

Beams usually sag between supports and hog over them, though this is not always the case. One, or two, layers of steel are allowed between supports, in BESS. A single layer is allowed over supports. Shear force is carried by stirrups, not by bent-up bars. The effect of shear force on local bond stress is checked as it is for slabs. Beams can only be rectangular in cross-section.

Columns carry a combination of thrust and biaxial bending moment. Shear force is not considered. Only rectangular columns are considered
and they are reinforced by one bar in each corner.

3.1.2. The choice of lower bound methods

The British Standard Code of Practice 114 for reinforced concrete\(^{(9)}\) allows the use of limit load or plastic design methods. These methods form the basis of the design philosophy adopted in BESS. In particular, extensive use is made of the Lower Bound Theorem. A statement of this theorem is 'If at any load factor \(\lambda\), it is possible to find a stress distribution in equilibrium with the applied loads and everywhere satisfying the yield condition, then \(\lambda\) is either equal to or less than the load factor at failure'\(^{(10)}\). Using the lower bound method has two main advantages. In the first place the resultant design is safe. In the second, design decisions can be taken about any one element when only the positions of the supports are known. In contrast, rigorous elastic analysis of a slab requires knowledge of the flexibility of its supporting beams. Such information is unnecessary when the lower bound theorem is used. This can be easily illustrated.

Consider in particular the span shown in Figure III.1.

---

\[ \text{Uniformly Distributed Load} \quad \text{BEAM} \]

**Figure III.1.**
The beam is subjected to a uniformly distributed load which causes the free bending moment diagram shown in Figure III.2a. The reactant moments of Figure III.2b are superposed on the free moments to produce the diagram shown in Figure III.2c. These reactant moments could be given any value and, by the lower bound theorem, the beam would stand provided it was able to withstand the resulting bending moments. The engineer is free to choose the reactant moments and may distribute them as he wishes. Clearly he can choose a pattern which minimises the maximum bending moment to be withstood by the beam. The selection of beam strength is therefore dependent upon selection of bending moment diagram. The beam may be supported by columns or other beams but its strength is not dependent on theirs. By contrast, in elastic design it is necessary to know their strength (or flexibility) before the analysis can take place.

**Free Bending Moments**

![Free Bending Moments](image1)

*Figure III.2a.*

**Reactant Bending Moments**

![Reactant Bending Moments](image2)

*Figure III.2b.*
Final Bending Moments

However, the method does have its disadvantages. A poor choice of reactant moments can result in a heavy structure, which cracks. In the beam example cracks would occur if all reactant moments were set at zero and accordingly the beam was not reinforced over the supports. Against this, if a bending moment diagram is "well balanced", such large crack-producing rotations do not occur. (Well balanced signifies that adjacent positive and negative bending moment maxima are made as nearly equal as possible.) Moreover the better balanced section will generally be the lighter, since for construction reasons reinforced concrete cross-section dimensions do not usually vary along a span. An implication of this is that the cross-section dimensions are determined by the largest moment, which is small in a well balanced span.

To summarise, use of the lower bound theorem has the following advantages:

a) The safety factor is at least unity.

b) The design process is quicker because it relies on simple bending moment distributions, not on element behaviour.

The disadvantage of the method is that if it is used ill-advisedly it can lead to heavy structures, which crack.
3.2. **THE LOADING**

3.2.1. *Introduction*

Elastic design involves examining the integral behaviour of a building and this requires a full analysis. In the iterative design process the number and size of steps is very great if one is modifying structural elements on the basis of analysis of integral behaviour. Full analysis is inappropriate at the initial stage of design and many approximate methods and rules have been developed in its stead. These methods are frequently unnecessarily inaccurate, since they ignore the basic properties of the structural elements.

Loads are carried by slabs, slabs by beams and beams by columns. The lower bound method allows the engineer to design these elements consecutively. As the loads are carried, so the design order is determined. In this method the engineer is defining instead of observing how the loads are carried.

However, the lower bound method could not be used by hand as it is in BESS. In defining how the loads are to be carried by one member, one defines the load falling on the supporting members. When the supporting members are designed, reference must be made to those members they support in order to find the actual loads. For example, this is simple for a beam which supports one slab, but not for a complex continuous beam which carries the load of, say, twelve slabs. It could be a day's work just to determine the loads carried by one member if its situation was very complex. Using a sophisticated model, a computer would typically produce such information in well under a second.

The situation is further complicated by the requirements of
BSCP114 that the following load cases be considered for beams and slabs:

a) Alternate spans loaded and all others unloaded.

b) Any two adjacent spans loaded and all others unloaded.

(In the event that a slab delivers some load to one span and some to others, the task of determining a 'worst case' by hand, becomes prohibitive.)

When a beam has been designed, both free and reactant moments are defined as well as the loads placed on the columns. Similarly, when slabs have been designed, the free moments on supporting beams are fixed. Once the depth is selected, the design of slabs depends on two choices. First the distribution of load in the two directions of reinforcement must be chosen. Second, the reactant moments must be chosen. The latter choice occurs after the former, which is discussed in the next section.

3.2.2. Distribution of loads in slabs

The behaviour of slabs which carry loads in two directions is very complex and even in elastic analysis it is necessary to make some simplifying assumptions. It is convenient to consider that \( N \% \) of the total load is carried in one direction with \( (100 - N)\% \) carried in the other. BSCP114 derives a rule for distribution by regarding the load as being supported by two sets of independent beams. The value of \( N \) is determined by equating the central deflections of these orthogonal beams. Using this theory for a square slab, the loads carried in the two directions are obviously equal. In highly loaded slabs, the method is inefficient, since one layer of steel has a greater lever arm than the other and consequently the concrete is not fully stressed in both directions. In BESS the percentage, \( N \), is calculated using the fact that one layer of steel runs over the other.
The formula used in BESS is now derived according to the Lower Bound Theorem. From BSCP114 the resistance moment of reinforced concrete, $M_r$, is given by:

$$M_r = P_{cb} \frac{b d_1^2}{4}$$

where

- $P_{cb}$ = permissible concrete compressive stress due to bending,
- $b$ = section width,
- $d_1$ = depth to tension steel.

In a balanced section, the steel and concrete give equal moment resistance. Also the resistance moment is at its maximum in both directions.

The resistance moments are given by

$$M_{rx} = P_{cb} \frac{b d_1 x^2}{4}$$

$$M_{ry} = P_{cb} \frac{b d_1 y^2}{4}$$

(wherex and y denote the transverse directions)

For strips of concrete of unit width we have

$$\gamma = \frac{M_{rx}}{M_{ry}} = \frac{d_1 x^2}{d_1 y^2}$$

(III.1)

where $\gamma$ is the ratio of resistance moments.

If the total load on the slab per unit area is $w$, and $aw$ is carried in the x-direction, then $(1 - a)w$ is carried in the Y-direction. Hence the applied moments are

$$M_x = aw \frac{\ell_x^2}{8}$$

(III.2)

$$M_y = (1 - a)w \frac{\ell_y^2}{8}$$

(III.3)

where $\ell_x$, $\ell_y$ are slab span lengths.
Further we may obtain the ratio of applied moments

\[ \frac{M_x}{M_y} = \frac{a}{1 - a} \left( \frac{a}{b_y} \right)^2 \]

Using \( M_x = M_{rx} \) and \( M_y = M_{ry} \) in equation III.1 gives the expression

\[ \gamma = \frac{a}{(1 - a)} \left( \frac{a}{b_y} \right)^2 \]

\[ = \frac{a}{(1 - a)} \cdot K \]

where \( \gamma = \left( \frac{\partial M_x}{\partial y} \right)^2 \) from III.1.

and \( K = \left( \frac{a}{b_y} \right)^2 \)

Hence

\[ a = \frac{\gamma}{K + \gamma} \quad (III.4) \]

The \( \alpha \) calculated in equation III.4 is an important feature of BESS's slab analysis and is referred to as the alpha-factor. It can be used to calculate the distribution of load on a slab so that the concrete is fully stressed, and is thus ideal for the heaviest loaded slabs - the most critical ones. The alpha-factor appears to be based, from the arithmetic, on the assumption that the slab is simply supported on all four sides, (see equations III.2 and III.3), but the essential fact is that whatever replaces the factor \( \delta \) should be the same for both equations. This is likely to be so for highly loaded slabs, where the reactant moments will be equal to half the applied moments, for best results, and if it is not the user can over-ride this default value and select his own.
3.2.3. Loads on beams and columns

The live load on beams arises from directly imposed loads, such as non-structural walls, and from slabs. Once the alpha-factor is set for the slabs on either side of a beam it only remains to specify the directly imposed loads. They are input to the program as uniformly distributed loads.

No loads are directly applied to columns. They all arise from other members.
3.3. THE TESTS OF MEMBERS

3.3.1. The limits of bending moment, shear and thrust

It is advantageous to use the same tests for beams and for slabs, since the forces and moments on slabs are a simplified case of those on beams. However, although they could be taken as columns with zero thrust, it is easier to test columns entirely separately. The actual process of testing is interwoven within the whole sizing process and could be approached from many different angles. The chosen approach was selected because it fits interaction.

For slabs and beams, the tests are applied after the bending moment diagram has been selected. Columns are treated differently. Each column's maximum capacity is determined before the sizing process commences. Then the loads placed on the columns are monitored continuously and the user is informed if he overloads any column.

The tests must cater for the worst combination of forces that may arise and they must also cater for all possible permutations. For sections in bending and shear the worst case is when both act together. However, the steel to resist varies between cross-sections. Either one or two layers of steel can be used to take tension. Either none or one layer of steel can be used for compression. If compression steel is allowed then so are stirrups, to resist shear. For sections in bending and thrust the worst case involves the application of biaxial bending moment together with thrust.

There are many points in the testing process at which failure can occur. The tests have been organised so that as much information as possible can be given back to the user concerning the cause of failure.
3.3.2. **Tests for cross-sections in bending and shear**

The algorithm which tests this is shown as a flow chart in Appendix A. However, the theory used is described in this section. Bending is treated first, followed by shear. These calculations are based on BSCP114 rules for bar details using the load-factor method.

The assumed stress distribution has the form shown in Figure III.3. The distribution is for working loads.

In the Figure, \( C_c \) is the compression force in the concrete, \( C_s \) is the compression force in the steel above the neutral axis, and \( T \) is the tension force in the steel below the neutral axis. They are each given by the equations below.

\[
C_c = \frac{2}{3} \cdot P_{cb} \cdot d_n \cdot b
\]
where \( p_{cb} \) is the permissible stress in concrete in compression due to bending.

\( d_n \) is the depth to the neutral axis as shown in the figure.

\( b \) is the breadth of the section.

\[ C_s = p_{sc} \cdot A_{sc} \]

where \( p_{sc} \) is the permissible stress in steel in compression.

\( A_{sc} \) is the area of compression steel

\[ T \cdot p_{st} \cdot A_{st} \]

where \( p_{st} \) is the permissible stress in steel in tension

\( A_{st} \) is the area of steel in tension

From CP114, there are two moments of resistance, one due to steel \((M_{rs})\) and the other due to concrete and compression steel \((M_r)\).

The moments are given by

\[ M_r = p_{cb} \cdot \frac{b \cdot d_1^2}{4} + A_{sc} \cdot p_{sc} \cdot (d_1 - d_2) \quad (III.5) \]

\[ M_{rs} = A_{st} \cdot p_{st} \cdot \ell_a \]

where \( \ell_a = d_1 - \frac{3}{4} \cdot \frac{A_{st} \cdot p_{st}}{b \cdot p_{cb}} \)

In the sizing operation, everything but the areas of steel \((A_{st}, A_{sc})\) and their positions \((d_1, d_2)\) are known. The positions are calculable from bar diameter. Thus for any bar, only the areas remain unknown. They are given by the following equations.

\[ A_{sc} = \left( M_r - \frac{p_{cb} \cdot b \cdot d_1^2}{4} \right) \div \left( p_{sc} \cdot (d_1 - d_2) \right) \quad (III.6) \]

\[ A_{st} = \frac{d_1 \cdot p_{st} - \sqrt{d_1^2 \cdot p_{st}^2 - \frac{3}{4} \cdot \frac{p_{st}^2 \cdot M_r}{b \cdot p_{cb}}}}{2 \left( \frac{3}{4} \cdot \frac{p_{st}^2}{b \cdot p_{cb}} \right)} \quad (III.7) \]
(The other solution to the quadratic is dismissed because it represents a state where the tension steel is in compression and the concrete is in tension while the section is very deep so the steel is at the top above the neutral axis.)

If two layers of tension steel are used, three assumptions are made. First, maximum compression steel is necessary and is fully stressed. Second, the concrete is fully stressed in compression. Third, the lower layer of tension steel is full and fully stressed. In this situation the resistance moment due to the lower layer of steel in tension is calculable. The resistance moment due to the compression steel is also calculable.

The applied moment can be considered to be resisted by two hypothetical beams acting together. One consists of the lower tension steel, the compression steel and concrete under stress $\sigma_1$. The other consists of the upper layer of tension steel and concrete under stress $\sigma_2$. The sum of the concrete stresses, $\sigma_1$ and $\sigma_2$, must not exceed the permissible stress, $p_{cb}$.

In both hypothetical beams the resistance moments due to compression zone and tension zone are equal. However, in the first, the resistance moment is known (it is that due to tension steel), and that due to compression steel is also known. Substituting $\sigma_1$ for $p_{cb}$ in equation III.5. and re-arranging, we have

$$\sigma_1 = (M_{rs} - A_{sc} \cdot p_{sc} \cdot (d_1 - d_2)) \cdot \frac{b \cdot d_1^2}{4}$$

But $\sigma_1 + \sigma_2 < p_{cb}$

so $\sigma_2 < p_{cb} - \sigma_1$

The second hypothetical beam is made of a single layer of steel, and
unassisted concrete in compression whose stress must not exceed $p_{cb} - \sigma_1$. Therefore if $p_{cb}$ is reset to the lower value of $p_{cb} - \sigma_1$, we can use the test outlined for a single layer of steel.

For shear, the theory is simpler. BSCP114 gives an equation for shear stress

$$q = \frac{Q}{b \cdot \ell_a}$$

where $Q = \text{gross shear force}$. The stress, $q$, must be less than four times the permissible shear stress. Whatever its value, shear must be resisted by steel, in primary members (beams). The shear resistance of steel stirrups is given as

$$Q = \frac{p_{st} \cdot A_w \cdot \ell_a}{S}$$

where $A_w$ is the area of the stirrups

$S$ is the spacing along the beam.

The maximum number of bars in any cross-section is easily calculable. $A_w$, the area, follows directly from the number and diameter. The only unknown is the spacing $S$. This can be calculated and checked against the code's requirements. It is worth noting that slabs supported by beams have little problem with shear stress. An unloaded slab (of any depth) would need to span thirty metres to become a problem even if made of weak concrete and spanning only in one direction.

There is no check for average bond stress. This is rarely a problem in the type of construction that BESS models. It is more a detailing problem than a critical design issue. In contrast, there are checks for local bond stress. The stress is calculated according to the formula in BSCP114 which is

$$\text{Local bond stress} = \frac{Q}{\ell_a}$$
where \( o \) is the sum of the perimeters of the bars in tensile reinforcement. This stress must be less than the permissible local bond stress.

The testing algorithm produces one of five verdicts. One verdict passes the cross-section if it is adequate. The other four signify different types of failure. They are designated as Imaginary, Local bond, Compression or Bars. The Imaginary failure occurs when equation III.7 produces an imaginary root, indicating gross undersizing of the member. Failure due to local bond is self-explanatory and indicates slight undersizing. Compression failure occurs when the resistance moment due to concrete and steel in compression is inadequate. Bars represents the failure which results when the test algorithm has tried all bar diameters but the necessary area of steel cannot be provided by any of them, due to inter-bar spacing requirements.

3.3.3. Handling cross-sections in bending and thrust

Figure III.4 shows the variation of moment with thrust for a column with high-yield as-rolled steel (in common use). It shows that, for this steel in this arrangement, the largest error one can make by approximating the relationship by a straight line between the intercept is 12%. This is typical of reinforced columns of rectangular type and though the error is much greater for mild-steel, mild-steel is used very seldom for main reinforcement. BESS approximates column behaviour in this way.

Furthermore, one is not really dealing with uniaxial bending but biaxial bending, and the relationship is not a line but a surface, usually known as the 'interaction' surface. A pictorial representation of this can be seen in Figure III.5. In plan the curves look like Figure
**Symmetrical High Yield Reinforcement**

**As-Rolled**

Theoretical curve

Assumed curve (straight line)

Differential $= 1.04 - 0.92 = 12\%$

$1.04 \times 0.01$

**Figure III**

- $A_{st}$, $A_{sc}$
- $d_1$, $d_2$
- $A_{ef} = A_{sc} = 0.5rbd$
- $P$: Load on column
- $M$: Bending moment about $E$
- $f_{ce}$: Permissible stress in concrete in direct compression
- $f_{ct} = 210$ N/mm$^2$ (20 mm dia bars & under)
- $f_{ct} > 210$ N/mm$^2$ (over 20 mm dia)
- Denotes part of graph to be used for bars over 20 mm diameter
Thrust

"X" Bending moment
"Y" Bending moment

FIGURE III.5

"X" Bending moment
"Y" Bending moment

FIGURE III.6
III. It can be seen from Figure that a straight line is also not far out as an approximation of this curve. How 'far out' varies from column to column. The important feature is that the straight line approximation conforms to the lower bound approach.

The interaction surface is therefore idealized as a flat-plane intercepting three independent axes (thrust, bending in x-axis plane, bending in y-axis plane). The intercepts are easy to calculate. They are pure thrust, and pure bending in each of two directions.

The equations used for the intercepts of the interaction surface on the axes representing pure bending are the same as already shown. The equation used to calculate the intercept for thrust is:

\[ F' = P_{oc} A_c + P_{sc} A_{sc} \]

where \( F' \) is permissible thrust (unmodified)

\( P_{oc} \) is permissible thrust in concrete in direct compression

\( A_c \) is the area of the column in cross-section

and the actual allowable thrust (\( F \)) is given by:

\[ F = K F' \]

where \( K = 1.5 - \frac{r}{30} \) for \( 15 < r < 33 \)

and \( K = 0.625 - \frac{r}{120} \) for \( 33 < r < 57 \)

in which \( r \) is the ratio of effective column length to least lateral dimension of column. Once the intercepts have been calculated for any column it is a simple matter to decide if a column is subjected to too much thrust or bending moment. As the loads are adjusted, the thrust
on the column varies. For each thrust the computer quickly calculates the moment and informs the user, if it is too great. In BESS one chooses the ratio between $M_x$ and $M_y$ (the resistance moments in either direction). Then given a thrust, two permissible moments are calculated and checked. In this way BESS checks columns and the method has the advantages that it is quick and of low error.

BESS idealizes all columns as though they have four bar reinforcement, which is generally adequate up to 750mm square columns. Columns rarely carry high bending moments so that this restriction is not very severe.
3.4. USE OF LOADING AND TEST PROCEDURES

3.4.1 Slabs

This section shows how BESS uses the load and test procedures in the sizing process. The operation is set out below.

a) Engineer selects a continuous chain of slabs, using the light-pen from a floor plan having set their sizes previously.

b) A check for continuity is carried out.

c) Data is extracted concerning dimensions and support moments.

d) \( \gamma \) and \( \alpha \) factors are calculated.

e) Load to be carried is calculated.

f) Loading case is entered by engineer (two adjacent or alternate spans loaded).

g) Corresponding free bending moment diagrams and reactant moments are displayed.

h) Engineer

i) chooses new load-case

cycled

or

ii) chooses new alpha-factors

round as

or

iii) alters support moments

often as

or

iv) submits set of slabs to test algorithm wanted.

(Section 9)

j) Test algorithm gives pass and fail verdicts.

Engineer cycles round h and j, or rejects calculations and exits, or accepts calculations whereupon the steel details calculated are stored together with the moments and the shear forces in the data structure and the routine exits.

(After exit new sizes can be chosen and the whole process restarted.)
### 3.4.2. Beams and Columns

These two elements are sized together because the moments are closely linked. Many of the load-cases which are critical for beams are also critical for columns. They are therefore handled more quickly together than separately. Their sizing process is set out below.

a) The engineer chooses a line of beams from a side elevation.

b) The computer
   i) checks they are in line
   ii) checks for "integrity of supports" (explained later).
   iii) finds those members (beams and slabs) which affect the chosen set and displays them in plan.
   iv) assumes all slabs are loaded, computes column loads
   v) checks to see if columns can support such loads. If not, it requests size alteration.
   vi) calculates intercepts for columns moment-thrust interactions
   vii) displays free and reactant bending moment diagrams.

c) The engineer may then
   i) choose which slabs are to be loaded
   or ii) alter reactant moments at joints
   or iii) submit the members to the test algorithm and accept or reject as for slabs.

Some additional facilities which are not essential to theory are discussed in Chapter 4.
The "integrity of supports" concerns beams which support other beams. If the supported beams have not been sized the load they place on the support beam cannot be calculated. Consequently, the support beam cannot be sized. In such a circumstance, the integrity of supports is inadequate, and the program prohibits sizing until integrity is achieved.
CHAPTER IV

USE AND VALUE OF INTERACTION USING BESS

4.1. INTRODUCTION
4.1.1. General
4.1.2. Logs and Tables - explanation
4.1.3. Decision Charts - explanation
4.1.4. Building used.
4.1.5. Timing

4.2. TOPOLOGY AND GEOMETRY
4.2.1. Introduction
4.2.2. Viewing and Examining
4.2.3. Setting sizes (geometry)

4.3. SIZING SLABS
4.3.1. Introduction
4.3.2. Description and Simple Example

4.4. SIZING BEAMS AND COLUMNS
4.4.1. Introduction to the process
4.4.2. Introduction to two examples
4.4.3. Simple example
4.4.4. Full example

4.5. SUMMARY
4.1. INTRODUCTION

4.1.1. General

In this chapter an account is given of experiments performed on the BESS system. In some circumstances it is easy to assess the effectiveness of the system. For example, the speed of design can be measured. It is much more difficult to assess the effectiveness of the system as an aid to obtaining a satisfactory design, because the personal judgement of the individual designer is involved.

The system was used to size a building of rather complex geometry. It is simple to present details of the final design together with the time and effort required. Much more difficult is to recount the progress of the design process. This has been attempted by keeping a LOG which records not only the actions and time durations but also impressions and motives for taking particular lines of action.

4.1.2. Logs and Tables

These are used to record and show the processes involved in examining and designing the building and to show the timing of these processes by being a diary of development. LOGS are grosser tabulations of raw data while TABLES present essential information, or digests. The LOGS were taken at the machine during actual sessions 'on-line' and tabulate the sequence of actions.

The following information is presented in columns:

a) ACTIVITY NUMBER

b) DESCRIPTION OF ACTIVITY
c) **ACTUAL TIME AT THE END OF THE ACTIVITY**

d) **ACTUAL DURATION OF ACTIVITY**

e) **ACTUAL DURATION OF TIME SPENT WAITING FOR THE MACHINE'S RESPONSE**

A distinction is made between (d) and (e) because (d) is the time spent in constructing a command to the machine together with the execution time (e). In order to convey the subjective experience, significant ideas and impressions were recorded during each LOG. A LOG is shown in Figure IV.7, which gives some idea of a session of BESS. The plots mentioned were produced by the digital plotter of the machine under the control of BESS and are shown in Figures IV.7 to IV.13. All of these will be discussed more fully later.

**TABLES** show data in a more refined form than that of **LOGS**. What is worth putting into a **TABLE** varies from LOG to LOG. In Figure 2, **IV.2**, TABLE 1 shows a comparison of the rates of picture display and plotting. This data affords evaluation of the performance of BESS from the user's view - his frustration and boredom in this case.

### 4.1.3. Decision charts

At any point in BESS there is a finite number of possible actions open to the user. Any activity has its own set of sequels. An understanding of the possibilities of each section of BESS is essential to the study of how it was actually used. For this reason **Decision charts** are included which indicate the possible paths through each section. They are so termed because the user must decide explicitly on the path he wishes to select.

Figure IV.4 illustrates the decisions to be taken inside the **VIEW**
LOG 1: VIEWING PLANS AND SECTIONS

Date: 17 February 1973

Notes:
- C/E indicates a pause to 'check and examine'
- W/N? indicates a pause to decide 'what next?'
- PLOT means plot the last view.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>TIME</th>
<th>DURATION</th>
<th>ELAPSED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR:MM:SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>0</td>
<td>BESS LOADING</td>
<td>16:00:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VIEW TOP FLOOR (4th)</td>
<td>.20</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>C/E: What's different from Ground Level?</td>
<td>.30</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>PLOT (Figure IV.13)</td>
<td>.50</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>VIEW GROUND FLOOR</td>
<td>1:10</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>C/E: OVERHANG + OUTSTAND</td>
<td>1:20</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>PLOT (Figure IV.12)</td>
<td>1:40</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>VIEW SECOND FLOOR</td>
<td>2:05</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>C/E: Outstand stops at 2nd floor - what effect does this have on beams? View sections</td>
<td>2:15</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PLOT (Figure IV.7)</td>
<td>2:40</td>
<td>25</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>VIEW X-Section at 'root' of outstand</td>
<td>3:10</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>C/E: Shows beams of different lengths all the way down. What happens to columns? - Very Varied Moment/Thrust ratios.</td>
<td>3:30</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PLOT (Figure IV.8)</td>
<td>4:00</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>ZOOM X-Section at 'root'</td>
<td>4:30</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>C/E: Ratios of spans are 3:4 &amp; 4:1!</td>
<td>5:15</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

So on lengths alone moments will be $\frac{16}{9}$ and 16 times each other on adjacent spans.
LOG 1 : VIEWING PLANS AND SECTIONS

Date : 17 February 1973

Notes :

C/E indicates a pause to 'check and examine'

W/N? indicates a pause to decide 'what next?'

PLOT means plot the last view

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR.MM.SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>15</td>
<td>PLOT (Figure IV.10)</td>
<td>5.45</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>VIEW Y-Section on North face of outstand</td>
<td>6.00</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>C/E: Large unbalanced span lengths once outstand starts, but higher up not bad so moment/thrust ratio not too bad.</td>
<td>6.15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PLOT (Figure IV.9)</td>
<td>6.30</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>19</td>
<td>ZOOM Y-Section on North face</td>
<td>6.45</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>C/E: Span length ratios 2:1:1 &amp; 1:1. So on lengths alone the moments are 6 times each other on adjacent spans. - Hope for high thrust.</td>
<td>7.10</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PLOT (Figure IV.11)</td>
<td>7.25</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td>W/N? Check OVERHANG size</td>
<td>7.55</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>VIEW X-'AC'</td>
<td>8.15</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>24</td>
<td>C/E: LARGE OVERHANG, but balanced by short internal beams which may take the reverse moment. Expect Y section the same.</td>
<td>8.35</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>VIEW Y-'AC'</td>
<td>08.55</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>26</td>
<td>C/E: As expected, but investigate corner sometime.</td>
<td>09.40</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

NB. Total wait time = 3 m 11 secs. = 33%
### VIEWING TIMES

#### TABLE 1

<table>
<thead>
<tr>
<th>Section</th>
<th>No. of Joints in Section</th>
<th>No. of Joints in Display</th>
<th>Time (seconds)</th>
<th>Joints/Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLANS (Floors)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUND</td>
<td>52</td>
<td>10</td>
<td>10</td>
<td>5.2</td>
</tr>
<tr>
<td>FOURTH</td>
<td>80</td>
<td>80</td>
<td>16</td>
<td>5.0</td>
</tr>
<tr>
<td>SECOND</td>
<td>83</td>
<td>83</td>
<td>18</td>
<td>4.6</td>
</tr>
<tr>
<td>PORTION of SECOND</td>
<td>83</td>
<td>33</td>
<td>18</td>
<td>1.8</td>
</tr>
<tr>
<td>X-SECTIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>71</td>
<td>71</td>
<td>16</td>
<td>4.4</td>
</tr>
<tr>
<td>PORTION of AC</td>
<td>71</td>
<td>42</td>
<td>16</td>
<td>2.9</td>
</tr>
<tr>
<td>Y-SECTIONS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>51</td>
<td>51</td>
<td>15</td>
<td>3.4</td>
</tr>
<tr>
<td>PORTION of AC</td>
<td>51</td>
<td>31</td>
<td>15</td>
<td>2.0</td>
</tr>
<tr>
<td>AO</td>
<td>33</td>
<td>33</td>
<td>6</td>
<td>5.5</td>
</tr>
<tr>
<td>PORTION of AO</td>
<td>33</td>
<td>20</td>
<td>6</td>
<td>3.3</td>
</tr>
</tbody>
</table>

For the full views, the average number of joints displayed per second = 4.7

**FIGURE IV.2**

*There is no Figure IV.3*
FIGURE IV.4
section. It is self-explanatory to some extent, though it is covered in
detail later. A few general points should be noted. At each point
marked with a cross, there are optional forward paths. To prevent the
charts becoming obscure no backward paths are shown unless essential.
At any cross the user may return to any previous cross. In each section
of the program, there is one starting point and one finishing point.

4.1.4. Building used

For all of the uses of BESS described in this chapter the same
building has been used. It was chosen early in BESS's life as a good
test of the system. Though an unrealistic building in many ways, in so
far as the features found would not occur in one building, each feature
is quite realistic in itself. Figure IV.5 shows the building in out­
line. It has four floors (ground, first, second and third) and a slab
roof. The ground floor is assumed to be adequately supported, since
BESS does not deal with foundations. There is an outstand on the East
face and an overhang on the West and South faces together with a 'void'
(hole) descending from the roof to the first floor.

The topology of this building is clearly muddled and this was
purposely done to stretch the system. The statistics are that it has
some 350 joints, 300 columns, 590 beams, and 170 slabs. (All figures
are taken to the nearest ten.)

4.1.5. Timing

It was intended to provide accurate central processor times
throughout the LOGS. There are two operating systems for the 4130.
One provides accurate times for usage of the central processor, the other does not. BESS was unfortunately too large to run under the former. For this reason times can only be estimated. It is shown in LOG 1 that for intensive use of BESS the proportion of real-time spent by the machine is 33%. Some further experiments were devised, using small programs, to achieve an approximate value for the proportion of that 33% spent by the CPU. The results are set out below.

The time involved in running BESS is the sum of four times:

\[ T = t_1 + t_2 + t_3 + t_4 \]  \hspace{1cm} (IV.1)

where

- \( t_1 \) is the CP activity involved in refreshing the display
- \( t_2 \) is the CP activity in updating the display file
  (e.g. deleting pictures, adding pictures etc.)
- \( t_3 \) is the "channel time" spent by the disk data transfer apparatus
- \( t_4 \) is the useful CP time

and \( T \) is the total real time for some machine activity. (No processing is done in parallel with disk seeks.)

A program was written which simply maintained a picture on the screen. The only CP time used was \( t_1 \) which was found to be 13.5% of real time \( T \).

A second program was written which updated the display file in the same manner as BESS's creation of views. The CP time involved is \( t_1 + t_2 \). From this \( t_2 \) was found to be negligible (0.05%).

A third program simulated BESS's drawing operation fully. It located data on disk, transferred it to core, abstracted data and created a picture. The only cause for CP time not to equal real time is that it spent time waiting for the disk transfers, time \( t_3 \). From this \( t_3 \) was found to be 52% of real time \( T \). Consequently \( t_4 \) must be 34.5% of \( T \).
In a typical BESS operation where the machine is involved for 33\% of real time, the central processor is involved in useful work (i.e. excluding display refresh) for \( \frac{33}{100} \times 34.5\% \) of real-time, or almost 12\%.

It should be borne in mind that picture drawing is, surprisingly, quite intensively "CPU-bound" by comparison with many operations, though not so much as some. Also, these figures arise from a period in which the machine was used intensively. Clearly one may expect fluctuations, but the author regards 42\% as surprisingly high. By comparison with other interactive computers it is a high figure.

The cost of operations using the equipment at Leicester University could be expected to be of the order of £30 - £40 per hour. It is by no means an ideal installation for this work. In Figure IV.6 a table is shown of the percentages of time spent on each activity.

FIGURE IV.6

CENTRAL PROCESSOR TIMES

<table>
<thead>
<tr>
<th>REAL TIME</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man</td>
<td>67</td>
</tr>
<tr>
<td>Machine</td>
<td>33</td>
</tr>
<tr>
<td>Comprising: CP useful</td>
<td>12</td>
</tr>
<tr>
<td>CP maintaining VDU</td>
<td>(4\frac{1}{2})</td>
</tr>
<tr>
<td>CP updating display file</td>
<td>-</td>
</tr>
<tr>
<td>Channel time</td>
<td>(16\frac{1}{2})</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>

(CP = central processor)
4.2. **TOPOLOGY AND GEOMETRY**

4.2.1. **Introduction**

In this section the term **TOPOLOGY** is used to describe the relationship of connected elements. **GEOMETRY** is used to describe the position and dimensions of the elements.

The user of BESS knows his building well before he comes to BESS's main interactive program, since he has created the model using BESS's input program by abstracting data from his drawings and then those of the architect. Moreover he will have discussed some detail with the architect already. The drawings he has seen so far, however, are very different to those produced by BESS. Architects and engineers show much incidental detail on their large scale drawing. The engineer later reduces his drawings to the bare minimum for design purposes. When using BESS this is done for him, automatically.

All the views presented by BESS are in design diagram form; they present very clear 'structural pictures' of the building. Indeed it is instructive to 'wander round' the building using BESS, for on such a journey the user extends his appreciation of the building as a structure. The scrutiny reveals a good deal of information for assimilation. Certain features prompt examination of other views which in their turn prompt again, in an iterative process. It can be useful to 'sample' a few bending moments and sizes but that will be discussed later.

Though BESS's views are purely two dimensional they convey information concerning members that pass through the plane being displayed. This is vital to the display. A beam-slab layout is nothing without details of the supports. Similarly a beam-column section is of little
value unless the transverse beams can be observed. There are four cases to be covered at each joint by the display program. It must distinguish between:

a) Member 'in front of' and 'behind' planes
b) Member 'in front' only
c) Member 'behind' only
d) None

Figures IV.7, IV.8, and IV.9 show the symbols used to distinguish these four kinds of joint from each other.

4.2.2. Viewing

The decision chart for VIEWing is shown in Figure IV.4. After selecting the VIEW option there are three decisions to take. First, the user must decide between using a default scale (SIMPLE), setting his own scale (ODD), and 'zooming' in onto a portion of building already displayed. (Under the zoom option the chosen portion is expanded to fill the available display area.) Second, the user chooses a floor plan or an X-section or a Y-section. The X-section runs parallel with the XZ axes, the Y-section runs parallel with the XZ axes. Third, he chooses either to type the information on the keyboard or to enter it using the lightpen. The information finally entered conveys to BESS the choice of plane and the part of that plane which is wanted.

The LOG in Figure IV.1 describes a typical examination of a building. This examination produced Figures IV.7 to IV.13. The second floor plan in Figure IV.7 was examined most intently and it shows the three main features of the building - the overhang, the void, the outstand.
BEAMS

BOTH

OTHER SIDE

THIS SIDE

NONE

FIGURE IV.7
BEAMS

• BOTH
• OTHER SIDE
+ THIS SIDE
+ NONE

X-SECTION "AL" ALONG O/S

FIGURE IV.8
FIGURE IV.8
BEAMS

- BOTH
- OTHER SIDE
- THIS SIDE
- NONE

FIGURE IV.9
Y-SECTION "AM" ALONG O/S

BEAMS

- BOTH
- OTHER SIDE
- THIS SIDE
- NONE

FIGURE 10.4
COLUMN

BEAMS

* BOTH
* OTHER SIDE
* THIS SIDE
* NONE

FIGURE IV.10
COLUMNS

BOTH

OTHER SIDE

THIS SIDE

NONE

DENOTES

A SLAB

FIGURE IV.11
<table>
<thead>
<tr>
<th>BOTH</th>
<th>OTHER SIDE</th>
<th>THIS SIDE</th>
<th>NONE</th>
</tr>
</thead>
</table>

A SLAB

FIGURE IV.12
external column

internal column

Uniformly distributed load

Bending Moments

Deflected SHAPE

FIGURE IV.14

FIGURE IV.13
a) "External" column

---

b) "Internal" column

---

\[ \tau \] = uniformly distributed load

---

**FIGURE IV.14**
Figures IV.8 and IV.9 show sections which are marked on top of Figure IV.7. Figures IV.10 and IV.11 are ZOOMs of Figures IV.8 and IV.9 respectively. Of the ZOOMs, Figure IV.11 affords an estimation of the scale of a problem which can easily be seen in Figure IV.8. The lengths of beams in Figure IV.11 on any one floor are very different. Disparate moments will arise, particularly around the column B-B which itself exists at the 'root' of the outstand (arrowed in Figures IV.7, IV.8 and IV.9).

These beam lengths can be seen in the floor plans, which show a further aggravation of the problem in that the spans of the slabs supported by the various beams also differ, so that the loads on these beams will diminish with their length.

The information just outlined was 'read' from several views generated by BESS. The times to do so are shown in LOG 1. This log was quite slow because of the plotting. Normally plots are rarely taken. If they are it is because the problem is too complex to be solved easily in an on-line session, and requires perusal to produce fresh ideas.

It is worthwhile listing the course of the examination in simple terms. After a look at the ground floor and top floor (4) Figures IV.12 and IV.13, the second floor showed the three prominent features. Engineering instinct was attracted by the outstand and the X-section (Figure IV.8) was checked to see the plane that is unchanged with height. Then the Y-section (Figure IV.9) was viewed to inspect column B-B. It can be seen that the moment/thrust ratio in column B-B is high above the second floor and low below it. It can therefore be expected that the outstand will be beneficial to the columns (though not necessarily to the beams).

All this took 92 minutes but it should be noted that column C-C, being similar to B-B was partially assessed. Other, less heavily
loaded columns which did not merit individual scrutiny have also been assessed. (Their assessment is simply that they will not be as difficult to deal with as columns B-B and C-C.)

The outermost (North-East) corner of the outstand would always be expected to be a problem. The columns at the corner receive large rotations from the long beams with no counter moment from the non-existent beams. Figure IV.14 shows the bending moment diagram for such columns (external), and, for comparison, that for normal columns (internal). The moments are normally lower in the internal columns.

The sequence necessary to arrive at the machine function DRAW takes five to ten seconds. It varies because of uncertainty of the best course of action. The time taken by the central processor is high, about ten to fifteen seconds for an average picture. The time to draw a section depends on the number of joints in the plane being viewed, not on the number which appear in the view.

The speed of BESS is not high when drawing sections and plans, as can be seen from TABLE 1. The data was split into a main file, and a directory file, it will be remembered from Chapter 3. This speeded the rate of drawing by a factor of at least five and, in cases by forty. However, for larger structures a three-tier system could speed it up by thirty or even forty times more.

Even so, the drawings come out at a rate which is not very much slower than a human being can follow (perhaps a factor of 3-4). What is frustrating is waiting for ZOOMed views. That would have to be altered in any viable system, if possible.
4.2.3. Setting sizes (geometry)

Before undertaking any testing of bending moments or comparisons of bending moment diagrams, the sizes of the members must be set. This section shows how the sizes are, in practice, conveyed to the program. The sizes fall into two groups. In one group there is slab depth and beam width. In the other there are beam depth and column dimensions. Beam width is put in the first group with slab depth because they affect the span of the slab.

The best point at which to set column dimensions is within the actual sizing operation. Beam depth is best set immediately prior to the sizing operation. For this reason only the process of setting slab depth and beam width is shown.

The Decision Chart for the data entry section is shown in Figure IV.15. The section is entitled ASK-PUT, because the user may ask what values exist in the data, or put new values in. If one interrogates the geometry of a member the X, Y and Z coordinates of the parent joint are displayed, together with the member's length, width and depth. The operation comprises two parts, one deals with individual members, the other with sets of members. In the former the user may interrogate or input, in the latter only input is allowed. The sets, or groups, are defined by the user. He must specify the plane in which they lie and the position and dimensions of a rectangle within the plane.

LOG 2 (Figure IV.16) shows the setting of slab depths and beam widths on the (roof) 4th floor. The live loading is high at 4kN/m². The slabs, except for the large group in the North-East (top right) can all be set at 150mm, which is near the practical minimum, but the heavier group needs at least 200mm. (This data is obtained by experience
FIGURE IV.15
LOG 2: SETTING SIZES

Date: 4 March 1973

Notes:

C/E indicates a pause to 'check and examine'

m/c means machine (i.e. the computer hardware)

There was so little 'elapsed' time in the entire LOG it was not recorded (less than 30 seconds total)

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BESS LOADING</td>
<td>22:05:45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VIEW 4th Floor Plan</td>
<td>6:10</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C/E</td>
<td>6:20</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Examine dimensions using ASK-PUT (span sizes)</td>
<td>7:40</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Obvious 'central' slab largest - can 250mm be used for this one? Others should all take 150mm, but if neighbours are 250mm the reverse moments may allow reduction at centre by increasing 'support' moments.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Set all slab depths to 150mm (Quicker this way in Real time)</td>
<td>8:45</td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Check a few to guard against machine error</td>
<td>8:55</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Set the N-East set to 200mm</td>
<td>9:30</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Check all (against m/c error)</td>
<td>9:50</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>COMMENT: Now for beams - 400mm wide for all beams - possibly not enough for some but will give a 'correction' to slab b.m.'s that may help.</td>
<td>9:50</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Set N-beams to 400mm wide</td>
<td>10:40</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Check a few (m/c error)</td>
<td>10:50</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Set E-beams to 400mm wide</td>
<td>11:40</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Check a few (m/c error)</td>
<td>11:50</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
LOG 2 :  SETTING SIZES

Date :  4 March 1973

Notes :

C/E indicates a pause to 'check and examine'
m/c means machine (i.e. the computer hardware)

There was so little 'elapsed' time in the entire LOG it was not recorded (less than 30 seconds total)

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR:MM:SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>13</td>
<td>Set All N-beams to 400mm deep</td>
<td>12:40</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Quicker in Real Time)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Set all E-beams to 400mm deep</td>
<td>13:30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Check some of each (m/c error)</td>
<td>13:50</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Set N-E group of N &amp; E beams to 600mm deep</td>
<td>17:50</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Check a few (m/c error)</td>
<td>18:00</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Total Time 12.15 secs.
and 'back of envelope' calculation).

A further feature is the setting of the whole floor followed by resetting some individual component sizes. This is done because the definition of a group is restricted to a rectangle and often it is quicker to reset a few than to set many individual members.

In twelve minutes the whole of the top floor was prepared for sizing - (slabs). It comprises 51 slabs, 130 beams. So 180 dimensions have been set (5½ seconds per dimension). The procedure was greatly slowed down by checks which were essential because of the unreliable behaviour of the electric typewriter. On any other computer the entering of data would be expected to be significantly quicker. It was not felt necessary to tabulate data from this section.
4.3. SIZING SLABS

4.3.1. Introduction

The Decision Chart for slab sizing is shown in Figure IV.17. The process involves a preliminary section, in which one chooses the slabs to be sized, and an operational section with nine options as shown. It will be recalled that the strength tests were outlined in the previous chapter. Most of the other options should be self-explanatory. The material properties are the various permissible stresses for concrete and steel. Using this option one may try stronger or weaker materials.

4.3.2. Description and simple example

Figure IV.25 shows a line of slabs whose identifying "0" is filled with a cross. This example illustrates the sizing procedure and Figures IV.18 to IV.23 are the bending moment diagrams taken during the process. The progress of the sizing is described in the LOG shown in Figure IV.24. In the LOG the spans are numbered from the left (1, 2 etc.). It is useful to read the LOG and examine the bending moment diagrams plotted in the process. The plots are in order as mentioned in the LOG, but activity plot has been omitted as it is so similar to the others and showed little of interest. Figure IV.25 duplicates Figure IV.13 but folds out for ease of reference.

The sizes of the slabs are as set in the ASK-FUT section. Using the default alpha-factor the spans just fail in compression. From the plan, which is displayed simultaneously with the bending moments, it appears that all spans except the first are almost square. In consequence their alpha-factors will be nearly 0.5. However, span 2 is small...
and can carry all its load in a N-S direction without disturbing the situation much. Accordingly, its alpha-factor was set to 1.0.

Figures IV.20 and IV.21 show the effect of the new alpha-factor. Figure IV.21 also shows a further modification (LOG activity 13) where the end span (5) has its alpha-factor altered, in order to reduce the moment on slab 4. The centre of slab 4 had previously failed and needed a slight reduction in bending moment, but support 4 was at its limit. After the alteration in span 5 all sections passed the test and the data was accepted.

The whole event took less than six minutes, including jotting down the LOG. The test algorithm takes at most 39 milliseconds for any section and more usually 20 or 15 msec. The verdicts are therefore almost instantaneous. For the fifteen spans centres and supports which were tested, the time lag was barely perceptible.
FIGURE IV.17

SIZE

SLABS

CHOOSE

CHAIN

interrogate bending moment diagram for actual values
alter reactant moments
choose alternate spans (load)
choose adjacent spans (load)
rescale or move b.m. diagram
alter any alpha-factor
obtain hard-copy of b.m. diagram
alter material properties
redisplay the bending moments

RETRY

ACCEPT

STORE DATA

EXIT
FIGURE IV.18

FIGURE IV.19
FIGURE IV.22

FIGURE IV.23
LOG 3 : SLAB SIZING

Date : 21 April 1973

Notes : REFER TO FIGURE IV.25.

Plot or PLOT means plot the last bending moment diagram.

TEST means subject all sections to the testing algorithm.

There was so little 'elapsed' time that it was not recorded (less than 6 seconds total) except for the earliest section of LOG.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR.MM.SS SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>1</td>
<td>LOAD BESS</td>
<td>00.00.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>VIEW 4th Floor</td>
<td>32 32 18</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Choose SIZE and slabs for sizing</td>
<td>52 20 0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Plot &quot;Alternate spans loaded&quot; pattern</td>
<td>60 8 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOTE: Moments at supports 2 &amp; 3 very small, but 3 will receive large rotation. Must make moment at support 3 &quot;as large as possible&quot;.</td>
<td>2.00 0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>VIEW: other alternate spans loaded and PLOT (Figure IV.19)</td>
<td>2.04 4 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOTE: The difference in moments across support 3 is now worse than before. Can ALPHA be altered? Considering plan - Yes to 1.0.</td>
<td>2.40 36 0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Alter ALPHA to 1.0 on slab 2</td>
<td>2.55 15 0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Now examine 4th slab loaded</td>
<td>3.00 5 0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Moment in slab 4 must be reduced. So try pattern (of support moments)</td>
<td>4.05 65</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Plot pattern (Figure IV.20)</td>
<td>4.07</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>TEST</td>
<td>4.09</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>One failure found in slab 4</td>
<td>4.11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>NOTE: Can ALPHA be raised on last slab (5)? - Considering plan - YES. Try 0.8 so as not to lose all its effects in transverse directions.</td>
<td>4.40</td>
<td></td>
</tr>
</tbody>
</table>
LOG 3: SLAB SIZING

Date: 21 April 1973

Notes: REFER TO FIGURE IV.25.

Plot or PLOT means plot the last bending moment diagram.

TEST means subject all sections to the testing algorithm.

There was so little 'elapsed' time that it was not recorded (less than 6 seconds total) except for the earliest section of LOG.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR.MM.SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>13</td>
<td>Set ALPHA to 0.8</td>
<td>4.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Increase moment at support No. 5 to allowable maximum</td>
<td>4.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Plot BM (Figure IV.25)</td>
<td>5.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Try other spans loaded</td>
<td>5.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Plot BM (Figure IV.21)</td>
<td>5.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>TEST</td>
<td>5.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>PASS</td>
<td>5.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Check 2 adjacent for reverse moments</td>
<td>5.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>PLOT (Figure IV.22)</td>
<td>5.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Check another 2 adjacent for reverse moments</td>
<td>5.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>PLOT (Figure IV.23)</td>
<td>5.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>O.K. TEST</td>
<td>5.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>STORE</td>
<td>5.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OFF</td>
<td>5.45</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COLUMNS

- BOTH
- OTHER SIDE
- THIS SIDE
- NONE

○ DENOTES

A SLAB

PLAN
4.4. **SIZING BEAMS AND COLUMNS**

4.4.1. **Introduction to the process**

Figure IV.26 shows the Decision Chart for beam and column sizing. As for slabs the engineer first chooses the members he wishes to size, but in this case from a side elevation. After the selection the computer performs three essential tasks before sizing can proceed. First, it checks the integrity of the supports and refuses to continue if the integrity is inadequate and remains uncorrected. Second, it displays a plan view of those slabs and beams which are involved in carrying loads to the beams and columns chosen. Third, it checks the dimensions of each column, (giving it those of the beams which cross over it if it has none) and asks for sizes if the dimensions give insufficient strength.

The plan view is useful for two reasons. It gives the essential visual confirmation that loads are being carried by the members intended to do so. Also, it allows the engineer to choose load cases by specifically selecting slabs to be subjected to live load. The check on column strengths is only rudimentary. It involves testing each column when all slabs carry their live loads. However, alteration of the reactant moments at either end of a beam results in modification of column loads.

Following the tasks described above, the program reaches the multiple choice section shown in the Decision Chart. The options are similar to those for slab sizing, the main difference being with the cross-section tests. Verdicts are given for five points on each beam (the quarter points and both ends), and double-banking of steel is offered to the engineer if the failure was of the local bond or the bars type. The other two failures are concerned with compression so extra tensile steel
would not help.

An option that is not in the sizing section but which is relevant involves the 'connection' of columns. All columns are regarded as connected unless specifically disconnected. A connected column is assumed to be capable of giving support, while a disconnected column is not. It can be seen from Figure IV.8 that if the outermost columns in the overhang are connected, the second floor beams must carry the entire weight of the overhang. Disconnecting the columns makes each floor cantilevered. If the columns are architecturally essential, they must remain and their strength removed. Using "dis-connection" and "re-connection" the effect of removing and replacing columns can also be examined, with minimal effort. Such examinations need care however. Columns are physical links and transmit deflections which are ignored in the analysis.

4.4.2. Introduction to two examples

Two examples now follow. The first deals with beam-column sizing and illustrates the detail of the process. The second (and last) example demonstrates the full scope of BESS, sizing all three members. Both examples have the same subject. They investigate the line of beams at the top of the building, shown in Figure IV.8. The beams support the slabs that were studied in the slab example. The first example concentrates upon beams and columns. It uncovers the fact that the strategy that was adopted for sizing slabs is detrimental to the sizes of the beams. The purpose of the second example is to resolve the conflict between the slabs and the beams. There are two matters which must be raised before the detailed discussion of the examples themselves. The first explains one aspect of the treatment of columns and beams. The
second raises the principal problems of sizing the subject members, before they are revealed by the sizing process.

The explanation concerns the bending moments at the joint between beams and columns. The sum of the bending moments in all members meeting at a joint must of course be zero. (The bending moment in any member is the sum of its free and reactant bending moments.) Therefore, if the sum of the bending moments in the beams is not zero, the remainder must be balanced by the columns. This remainder is referred to as the 'discontinuous moment'. In Figure IV.27, there are three bending moment diagrams. The first and second diagrams concern joints with non-zero discontinuous moments. The third diagram concerns a joint with a zero discontinuous moment. In addition, the first and third diagrams result from beams which are supported at both ends, whereas the second diagram results from the introduction of a cantilevered beam on one side of the joint.

Turning now to the second matter the principal problems of sizing the subject members can be deduced from the plan shown in Figure IV.13 and the elevation shown in Figure IV.8. These diagrams are repeated on a single page (Figure IV.40) behind the examples. The Figure folds out for ease of reference, and the encircled numbers are used as references in the text. The lengths of the beams are in the ratio 2:1:3:4:4:8:1:1. If they did not support slabs and all had the same cross-section, the free bending moments would be in the ratio 4:1:1:1:1:1:1 (to the nearest integer). The smallest moment would be about 5% of the largest. The second beam receives no load from its slab because the alpha-factor was changed in the slab example. All the others do carry loads, however, and the areas of the slabs vary widely. Consequently the free bending moments, and hence the reactant moments, can differ by a factor of one hundred. In
this situation, if the column is to carry a large discontinuous moment, then the resulting columns may be too bulky. In practice a decision must be made. The applied moments can be minimised and the columns given small dimensions. Alternatively, the columns can be subjected to high bending moments and their size increased greatly. The full solution of this problem involves examining the whole elevation from top to bottom, to account for the differing moment/thrust ratios. However, these examples are intended to show the scope and capacity of BESS, and not necessarily to find the best solution to this problem. Documentation of the solution would be very bulky, and in any case it would also be too specific to this one building to be of scientific value. The decision was taken, therefore, to examine only the situation where columns are small and take little moment.

4.4.3. Simple example

Secondary detail was removed from this example in order to convey the scope of the sizing procedure. Only the fourth beam was considered to carry live load, so that there were only two load cases. However, the effects of the disconnection of two columns provided plenty of material for discussion. LOG 4 describes the process.

The side elevation shown in Figure IV.4.0 was displayed and the sizing routine entered. The first two bending moment diagrams (Figures IV.28 and IV.29) show that the estimates made in the previous section were correct and that the arrangement of slabs and beams produces widely differing bending moments. The columns were only able to withstand a small moment as they were 0.4 metres square. The reactant bending moments at joint 5 were left at a low value in order to avoid placing a
large reverse moment on the fifth beam. This kind of moment on such a short beam could result in undesirably rapid changes of curvature and in an upthrust on column 6. The upthrust would not be bad, of itself, but all floors are similar and if they were all sized in the same manner the collective upthrust would have to be accounted for at the ground floor.

The effect of a substantial increase in column sizes is illustrated in Figure IV.32. In this case the allowable discontinuous moment is increased to 56 \( \text{kNm} \) by increasing the dimensions of the column from 0.4 to 0.6 metres square. It can be seen that the discontinuous moment is still small by comparison with the free and reactant bending moments. Consequently it is uneconomic to expect significant discontinuous moments to be carried by the columns. This conclusion could have been reached by an engineer in an ordinary drawing office. However, using BESS it took only ten seconds and in that time a wide range of values (of discontinuous moment) was covered, not just 56 \( \text{kNm} \). In addition, the display is precise and does not make mistakes.

For interest, a test was made to see how the beams were standing up to their bending moments and shear forces. From the test it became apparent that the beams were probably too small. Nonetheless the examination continued, and the first column was disconnected to make the necessary cantilever. The size of reverse moment, together with its distinct advantages with regard to altering reactant moments, prompted the question in activity 21 in the LOG. The penultimate column was also disconnected, as a result. The remainder of the LOG throws up two major findings that contrast with each other. The short beam (2) adjacent to the overhang is beneficial to the structure, whereas the short beams 5 and 6 would be better if replaced by one beam. Beam 2 divides the distance between columns 2 and 4 into two spans, each of which
LOG 4: SIZING BEAMS AND COLUMNS

Date: 19 May 1973

Notes: Plot or PLOT means plot the last bending moment diagram. Test or TEST means subject all sections to testing algorithm. 'Elapsed' times were only included where they give new information.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BESS LOADING</td>
<td>09.05.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VIEW X-section AL</td>
<td>09.05.30</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ENTER SIZING ROUTINE - choose nodes</td>
<td>09.05.35</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Support checks &amp; picture drawn</td>
<td>.50</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Column checking</td>
<td>6.05</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>Display of Bending moment diagram</td>
<td>6.15</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Plot (all BEAMS unloaded) (Figure IV.28)</td>
<td>6.20</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Load BEAM4</td>
<td>6.25</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Redisplay bending moment</td>
<td>6.35</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>PLOT (Figure IV.29)</td>
<td>6.40</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NOTE: BEAM2 BM barely visible. Alter support moments between BEAM3, BEAM4.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Moments altered</td>
<td>6.50</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>NOTE: Column takes virtually no bending moment (expect column to take 15KNm but beam moments 500KNm - so not surprising)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PLOT (Figure IV.30)</td>
<td>6.55</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>Alter moments BEAM4 - BEAM5</td>
<td>7.05</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>NOTE: Little to gain - again column takes very little, and to avoid too many changes of curvature moments must be kept small.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOG 4: SIZING BEAMS AND COLUMNS

Date: 19 May 1973

Notes: Plot or PLOT means plot the last bending moment diagram.
Test or TEST means subject all sections to testing algorithm.
'Elapsed' times were only included where they give new information.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>PLOT (Figure IV.31)</td>
<td>7:10</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Try increasing support moment at left end of BEAM4</td>
<td>7:20</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE</strong>: Column moment is violated but this tests what improvement can be expected from larger column (say .7m instead of .4m)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>PLOT (Figure IV.32)</td>
<td>7:25</td>
<td>.5</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>Unload BEAM4 &amp; redisplay</td>
<td>7:35</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>PLOT (Figure IV.33)</td>
<td>7:40</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>Test - failures: refuse double-banking and exit without storage</td>
<td>8:00</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>19</td>
<td>Disconnect first column (cantilever)</td>
<td>8:10</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>20</td>
<td>Re-enter sizing exactly as before</td>
<td>8:55</td>
<td>45</td>
<td>40</td>
</tr>
<tr>
<td>21</td>
<td>PLOT (Figure IV.34)</td>
<td>9:00</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>NOTE</strong>: Looks useful. If large 'reverse' moment thrown onto BEAM2 then BEAM3 much improved. However, why not try penultimate column too?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Exit (no test)</td>
<td>9:05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Disconnect penultimate column</td>
<td>9:15</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Re-enter sizing exactly as before</td>
<td>10:00</td>
<td>45</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>PLOT (Figure IV.35)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOG 4 : SIZING BEAMS AND COLUMNS

Date : 19 May 1973

Notes : Plot or PLOT means plot the last bending moment diagram.
Test or TEST means subject all sections to testing algorithm.
'Elapsed' times were only included where they give new
information.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Increase one end of BEAM2 and BEAM4 -</td>
<td>10.15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BEAM3 joint</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOTE: BEAM2 shear violation near</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>PLOT (Figure IV.36)</td>
<td>10.20</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Increase both sides of BEAM2 - BEAM3 to</td>
<td>11.20</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>match</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>PLOT (Figure IV.37)</td>
<td>11.25</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NOTE: Was confused by BEAM1 - BEAM2 -</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BEAM3. Applied and reactant moments looks</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>odd but is just the reversal of gradient</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>of lines of reactant moment. Sketch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>helped - remember in future for</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cantilever.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>TEST</td>
<td>11.30</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>31</td>
<td>Examine verdicts: Some failures. Don't</td>
<td>11.55</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>double bank, just leave it - it needs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>checking in transverse direction before</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>precise work is done.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Exit</td>
<td>12.00</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
carries a significantly smaller bending moment than a single span would have carried in their place. (A reduction of about 35% in free bending moments resulted from using two beams, judging from span lengths alone.) There was also an important reduction in the shear force imposed on the beams. In contrast, the short beams 5 and 6 cause the reactant moment at joint 5 to be smaller than its optimum value, by keeping the free bending moments at a lower value than those which would result from using a single beam.

The decisive result of this example is that column 6 should be removed and more of the slab load carried by beams 5 and 6. However, the decisions taken in the slab sizing operation directly contradict this. The alteration of the alpha-factor in the slabs which are supported (in part) by beams 5 and 6 removed weight from those beams rather than adding it. The incompatibility of these decisions is the subject of the next example, which illustrates how the problems were resolved.

4.4.4. Full example

In finding a solution to the problem revealed by the simple example, the sizing operation involved much of the top floor, even though it was only concerned with the right-hand end. Much more material is covered in this than in the previous examples. No plots were taken because they would have been too numerous, the LOG on its own simplifies presentation of a process which could otherwise become encumbered with detail. Individual actions have been described earlier in the chapter, consequently the LOG steps are larger than before and include greater discussion. Since there is no discussion outside the LOG, as there was previously, essential information must be set out first. This information concerns
the LOG itself and the method of testing double-layered tension steel.

It was assumed throughout the LOG that the outline of the structure could not be changed. The LOG, 5, uses the greek alphabet to identify slabs and Figure IV.40 shows the relation between the letters and slabs. (Alpha is not used in order to avoid confusion with the alpha-factor.)

The description of beams within the model only allows double-banked steel in either the top or the bottom of beams, not in both. (This restriction ensures that the percentage of any member's cross-sectional area which is steel remains below practical limits.) The testing algorithm is always applied to the loadcase that was last chosen. It is not possible to define any single loadcase as the worst case, so the designer has to use his judgement. Consequently, care is necessary in order to ensure that positive and negative bending moments do not both need steel to be double-banked.

The results of the process described in the LOG fall into three parts, the time of duration, the size alterations and the effects on the bending moments of the decisions taken. These three constituent parts of the results are now described.

A large proportion of the top floor was covered in sixty-three minutes. The time to write the LOG was included in the recorded times. The writing was helpful in crystallising the problems and their possible solutions, and is a useful feature of the operation of BESS.

The beams started grossly undersized and had to be enlarged twice. The slabs had already been accepted as satisfactory but were also enlarged, in depth, as a result of the conflict between beams and slabs. The area of the columns had more than doubled by the end of the operation. Some of the inaccuracy of initial sizes could have been avoided by more extensive use of rough calculations. However BESS is more reliable,
quicker, and reveals as much as hand calculations.

In Figure IV.41 three diagrams show the effective improvement, in bending moments along the span of beams, that was achieved during the sizing process. The improvement cannot be measured directly because the increases that were found to be necessary in cross-section incurred corresponding increases in free bending moments. The second diagram shows the bending moments that resulted from the operation and which correspond to the final sizes of the members. The first diagram shows the bending moments which would arise from the initial attempt at sizing if each member were given its final dimensions. The third diagram shows a direct comparison of the first two diagrams.
LOG 5 : SIZING ALL MEMBERS

Date : 23 May 1973

Notes : REFER TO FIGURE IV.40.

TEST or Test means subject all sections to the testing algorithm.

Since the intervals of time are so much larger here, only the 'finish'
time of each activity was recorded.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR.MM.SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
<tr>
<td>0</td>
<td>BESS LOADING</td>
<td>04.30.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>VIEW 4th Floor</td>
<td>04.30.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SIZE TOP LINE OF SLABS (β - γ) in East-West direction. Set Alpha-factors at East end to 1.0 (E - W). TEST - PASS</td>
<td>04.32.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>SIZE MIDDLE LINE OF SLABS (Δ - ε)</td>
<td>04.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alpha-factors: ε - use default for economy, Δ - set to 0,(E - W) because the well gives zero moment. TEST - PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>SIZE θ - φ SLABS</td>
<td>04.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use default alpha-factors TEST - PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SIZE π - σ SLABS</td>
<td>04.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alpha-factors set to 1.0 (E - W) to reduce cantilever moment.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>SIZE σ - γ SLABS</td>
<td>04.42.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alpha-factor of λ set to 1.0 (N - S). Moment diagram now nicely set out as before at southern end but very poor at northern end, because γ has no applied moment in N - S direction. Will have to accept large rotation of beam between ε - γ because alpha-factor cannot be altered since the wall on the east of ε leaves ε virtually simply supported in the E - W direction. TEST - FAILURE IN ε - should have expected it.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EXIT
LOG 5: SIZING ALL MEMBERS

Date: 23 May 1973

Notes: REFER TO FIGURE IV.40.

TEST or Test means subject all sections to the testing algorithm.

Since the intervals of time are so much larger, here, only the 'finish' time of each activity was recorded.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>ENTER ASK-PUT</td>
<td>04:43:30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>slabs in N-E are .2m deep so increase to .3m (SLABS y, ε, φ, θ)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Re-do steps 2, 3, 4 for new depth</td>
<td>04:44:30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Repeat 2 - TEST PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Repeat 3 - TEST PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Repeat 4 - TEST PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Increased reactant moments to balance larger applied moments.</td>
<td>04:46:30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Re-do step 6 as before</td>
<td>04:50</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST - PASS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXIT and Store</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>VIEW X &quot;AI&quot; section</td>
<td>04:51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ENTER BEAM - COLUMN SIZING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Columns still 0.4m square. Still adequate for thrust but will be useless for discontinuous moment. Much better moment at north end - can reduce largest moment significantly. (Disconnecting column 'C' would give excellent results. Remember, though, that 'C' would have to be physically disconnected from the joint to mirror analysis.)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TEST - FAILURES (some severe)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accept all double-banking.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Still failures - beam too small</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXIT</td>
<td>04:54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Enter ASK-PUT, find present sizes</td>
<td>04:57</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beam depths .6m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beam widths .4m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Columns .4m x .4m</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To make columns wider is attractive, but beams would have to be the same width to avoid awful form work construction (and the</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOG 5: SIZING ALL MEMBERS

Date: 23 May 1973

Notes: REFER TO FIGURE IV.40.

TEST or Test means subject all sections to the testing algorithm.

Since the intervals of time are so much larger, here, only the 'finish' time of each activity was recorded.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Need to design each joint specially to cope with a beam narrower than column to ensure full moment transfer from beams to columns.</td>
<td>05.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To make beams deeper is possible but there is only 2.15m (7'2&quot;) headroom anyway.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Try beam width .4m (ditto column) and depth .8m. Keep columns the same size.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Leave out slab resizing as beam widths are unaltered. RE-ENTER-BEAM - COLUMN section: BM diagram is very similar with slight increases in applied moments. Alter reactant moments to suit. TEST - FAIL (none severe) Try double-banking. TEST - PASS except a few. Solution would undoubtedly be achieved by making beams .6m wide and removing column 6.</td>
<td>05.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The position is now that all the transverse beams must be sized (those running E - W). So far their reactant moments have been taken as zero by the program when calculating their contribution to column thrust. As this will not be so they must be 'sized' before the final choice of beam dimensions. Note that they will, after sizing, probably give a smaller contribution to column thrust because of the likely reactant moments (high at West end, zero at East end). Bear in mind also that the beams between slabs ε and φ will very likely be widened to .6m as they are somewhat similar to the end beams under consideration.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LOG 5: SIZING ALL MEMBERS

Date: 23 May 1973

Notes: REFER TO FIGURE IV.40.

TEST or Test means subject all sections to the testing algorithm.

Since the intervals of time are so much larger, here, only the 'finish' time of each activity was recorded.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HR. MM. SS</td>
<td>SECONDS</td>
<td>SECONDS</td>
</tr>
</tbody>
</table>

NEXT PHASE:
Increase beam widths to .6m generally in the North-East section, and depths to .8m. Then size E - W beams at their Eastern ends (i.e., do not become involved in the overhang). Then return to the original set of beams.

N.B.
Size the beam which runs N - S between \( \theta - \phi \) slabs to better determine its reactant moments, but while doing ignore column thrust - leave till later. Postpone the short and the non-continuous beams which have less effect. (Short beams have small or zero reactant moments; non-continuous have zero reactant moments).

15 SIZE BEAM BETWEEN SLABS \( \theta - \phi \) 05.10
16 SIZE TRANSVERSE BEAMS 05.26
17 RE-ENTER SIZING OF ORIGINAL SET OF BEAMS AND COLUMNS - BENDING MOMENT DIAGRAM IS STILL THE SAME, VERY LARGELY (slightly changed in proportion).
Columns now take a slight moment, still barely perceptible.
Alter reactant moments
TEST - a few failures
Double-bank
TEST - all PASS

EXIT and Store 05.33

Notes:
Lower floors will be less of a problem as the column dimensions must (anyway) increase so beam widths can be greater. The
LOG 5: SIZING ALL MEMBERS

Date: 23 May 1973

Notes: REFER TO FIGURE IV.40.
TEST or Test means subject all sections to the testing algorithm.
Since the intervals of time are so much larger, here, only the 'finish' time of each activity was recorded.

<table>
<thead>
<tr>
<th>No.</th>
<th>DESCRIPTION OF ACTIVITY</th>
<th>FINISH TIME</th>
<th>DURATION TIME</th>
<th>ELAPSED TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>columns will carry no moment at any level. All that remains in this floor is treatment of the overhang, on South-West corner particularly.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE IV.39
Sheet No. 5
4.5. **SUMMARY**

This chapter attempted to demonstrate the usefulness of BESS. Actual examples of its use in practical situations were discussed. A full test of the entire building was not undertaken for reasons of time and space involved in both the test and its discussion. The times increased when all three members were sized, as the third example showed. Two facts arise from the third example. All the slabs on the fourth floor may be considered to have been sized, since one of each dimension has been sized. The beams have also been sized, but with a few exceptions. The complex in the bottom left-hand corner requires careful examination. In addition the beam that was sized from the overhang was not typical. It was the end beam and carried only half of the slab weight that the others carry (except for the other end beam, of course). There are one or two other situations in which slight differences occur, around the floor. For instance, though all of the overhang beams are balanced by short internal beams, not all of the internal beams are continuous with other beams.

Provided that these differences do not affect the floor too radically it may be stated that 106 beams and 42 slabs were effectively sized in sixty-three minutes. It is reasonable to suppose that the complex corner of the overhang together with the other differences would not take more than another hour. In any event, therefore, the entire building could be expected to be sized in ten hours. Allowing for the fact that each plane is somewhat similar, the time required for sizing the whole building might be four to five hours. At the charging rates mentioned in Section 4.1.5 the building would be sized, therefore, for 30 to 50p per column. On more modern equipment this cost would be expected to be
significantly less. The cost of the engineer is not included in the figures just given, however, and would become more significant if the costs dropped greatly.
CHAPTER V

CRITICISM OF SYSTEM

5.1. DEFICIENCIES OF FORMER PROGRAMS

5.2. AIMS OF BESS

5.3. EVALUATION OF BESS

5.3.1. Effectiveness in design

5.3.2. Improvements and fulfilment of aims

5.3.3. Costs
5.1. **DEFICIENCIES OF FORMER PROGRAMS**

In Chapter 1, the points were made that all use of computers is interactive, and that there is a relationship between the time of duration and the results of interaction stages. The results produced by a program seldom fulfil the exact requirements of the user. Additionally, the choice of method for a program does not always coincide with the user's own preference. The engineer bases his decision to use the computer on the time-result relationship, the adequacy of the results, and the sufficiency of the method. These factors are related to each other in the process of design.

In the early stages of a design, the engineer explores many very different possibilities (for example, the use of concrete against steel, or ribbed slabs against flat slabs). Extensive analyses have little value at such a time, because they involve large outputs. The need is for approximate calculations which allow quick assessment of different solutions. Batch processing is inappropriate to this need because of its inherent time penalties. In the later stages, design is involved less with making decisions and more with determining unknown quantities (for example, the details of steel bars in reinforced concrete). The time-result relationship of batch processing is compatible with this kind of work. Even so, the results are not necessarily adequate. At all stages of design, designers require confirmation that their intentions are being carried out. Such confirmation is often best given visually, but graphical output from batch programs carries extra time penalties. Batch output is often inadequate, therefore, for the designer's requirements. Moreover, output is strongly linked to the method, and the time penalties of batch affect the methods used in the programs. The
programmer attempts to counterbalance the slowness of a program by increasing its production, typically replacing the obvious method with a more powerful one. (For example, finite elements might replace more traditional methods in a plane frame program.) Such replacement commonly has adverse effects on the sufficiency of the method, which, in turn, adversely affects the results. Oversufficiency can be as bad as insufficiency.

A system such as GENESYS is an attempt to aid production of programs, and to cope with the problems outlined. Selective output obviously helps, but it cannot overcome the extra time needed for graphical output, without the use of very expensive microfilm plotters or graphics terminals. (The latter are unsuited to such systems because of poor response times.) Apart from these, systems have two inherent disadvantages. Systems are based upon batch processing techniques, since data processing has always been heavily batch oriented. Also, they limit the modelling capabilities of their subsystems by basing the facilities on unified schemes for handling data. Each problem needs its own model, and unified schemes constrain each model's features. The constraint on modelling is automatically a constraint on program or subsystem structure. Consequently the programs are often awkward, and their awkwardness is reflected by inflexibility and slowness in operation.

To sum up, programs in the past have been inherently unsuited to the early stages of design because they have been batch oriented and have not involved graphical output. Further, there is little hope of non-graphical, batch programs ever being useful since they have poor time-result relationships, produce inadequate output, and use inappropriate methods.
5.2. **AIMS OF BESS**

The initial study had four principal aims. The first was to employ an interactive, graphics terminal (in order to avoid one of the major pitfalls of previous programs). The second was to choose a subject commonly encountered in the design office. The third was to provide a program which would be useful in the early stages of design. The fourth was to choose a method of analysis suited to the subject, but also consistent with the other aims. BESS was produced in order to fulfil these aims, and naturally shares them.

The subject chosen is the reinforced concrete building. The relative structural simplicity of buildings (compared, for example, with ships), allowed the selection of a less sophisticated design method. Two advantages of such methods are that they require fewer complex calculations, and involve the designer to a greater degree than the more sophisticated methods. The fewer calculations it has to undertake, the quicker the machine responds. The more the designer is involved, the greater is his feel for the structure he designs. These advantages are a good base for providing help in the early design stages. They allow rapid assessment of different design solutions.

The method chosen is based on the lower bound theorem, which has the major advantage of being intrinsically safe. The method itself also gives the engineer a greater control over the structural behaviour of the building than elastic methods. (He defines where loads must be carried, instead of simply observing.) Greater control of the kind which forces the designer to take all decisions is tedious, and it was intended to avoid this trap by using default values wherever possible. The benefit of default values is that the designer knows that a sensible decision will be taken unless he specifically over-rides the program.
5.3. EVALUATION OF BESS

5.3.1. Effectiveness in design

The effectiveness of BESS is measured by its speed and the quality of the final design. Though speed is easily measured, quality cannot be fully assessed without extensive testing, by engineers used to other methods. However, the design situation which was examined in the last chapter affords some discussion of quality.

In the course of the design, many sizes were changed, with corresponding changes in bending moments. These changes in bending moments cloud the effects of design decisions, such as the alterations of alpha-factors. Consequently, to understand the effects of design decisions, the situation needs to be examined as though the sizes never altered. Figure IV.41 shows the relative changes in bending moments along the line of beams that were the principal study. (The members are assumed to have their final sizes.) The moments were reduced, in the best case, to 70% of their initial values. During the operation, the sizes of the beams were increased by a factor of two, those of the slabs by 50%, and those of the columns by 25%. These are striking alterations but were achieved in a short time. The same operation carried out by hand would either take a good deal longer, or be much less accurate. A practising engineer would have to achieve the same results in five to seven working days, in order to compete with BESS on cost. In the absence of experimental evidence, it is a matter of opinion as to which is better or worse.
5.3.2. Improvements and fulfilment of aims

BESS did not fulfil all the original aims. The principal faults are the slowness of picture drawing and of data input. Both could easily be improved in a rebuilt system. Improvements to BESS lie in one of two fields. They are either engineering extensions or improvements on the programming side. The ability to handle different types of construction (such as flat slab-column, ribbed or coffered slabs etc.) is clearly needed. The additional ability to handle members such as walls and stairs would be useful, but not essential. In general, walls are treated separately from the rest of the structure, and particularly when used to resist shear force. However, an integrated system such as BESS would possibly derive greater benefit from their inclusion than the present hand-based methods.

The improvements in computing techniques fall into two groups. The first is composed of those which could be effected on the ICL 4130 (or any machine), and the second, of those which need a specific machine.

A major need is for extended 'windowing' facilities. At present, bending moment diagrams are contained within the limits of the screen. This would cramp the picture for a long span of beams or slabs. However, the user could 'move around' the diagram if a window were provided (just as he moves around the views of the building). The modelling facilities could usefully be extended so that members could be added or removed. Finally, a 'repeat' facility would allow the user to copy data from one member to another. At present, all members must be sized explicitly, even if they are identical to a member which has already been sized. The facility to overcome this would save much time.

With more modern hardware, BESS could be greatly enhanced. An
ideal configuration for BESS might be:

- FDP 11/45 processor with 32K of store and 10 Megawords on disk
- Vector General Display with hardware capacity to handle three-dimensional pictures (rotate, scale and translate them).

The windowing function as outlined for any machine, would be performed a great deal more quickly on this configuration. BESS would be able to handle buildings of up to 6000 columns, on such a machine, and all operations would be speeded up. A very significant improvement would be made in cost, which is discussed in the next section, but perhaps the most important benefit would be in convenience. The machine response-time would drop for the graphical functions, which are trivial to the engineer. Moreover, because the machine is considerably cheaper than the 4130, there would be less urgency, less need to use as little time as possible. The quicker response and decreased urgency would allow the user to concentrate more, and to take more rational decisions.

5.3.3. Costs

As mentioned in Chapter IV, the cost of using BESS for sizing the 350 column building would lie between 30 and 50 pence per column. The engineer's time has not been included since he is only occupied for a few hours. The preparation time and the cost of the initial run to create the model were not included either. It took the author about one day to generate the model. Fifteen minutes of machine time were also
used. These costs would add another 8p to the cost per column. It is difficult to compare this with figures for hand calculations, since extrapolation from these figures is uncertain. In a building with greater floor area than the one studied, the process might have taken proportionately more time if there were greater interaction of the members. Alternatively it might be proportionately quicker if the extra detail were simple and could be dealt with easily. It is dangerous to try to extrapolate too much from one building, designed by one person.

It is certain, however, that costs would be considerably lower on the ideal configuration outlined in the previous section. Ideal equipment would provide better facilities and also allow quicker operation. The annual cost of the configuration would be about £12,000 or £15,000, allowing it to depreciate over a decade. Assuming that the work existed for it to be busy for forty hours per week, it could be used to size about 12,000 columns per month. Machine costs would therefore be about 8p per column (£28 for the 350 column building). This figure is not precise, of course, being based on only one design. Nonetheless it shows that considerable benefits will accrue if BESS can be produced - and BESS can be produced.
REFERENCES


5. MILLER, C.I., General Description of ICES COGO, Massachusetts Institute of Technology, 1966.


APPENDIX A

An algorithm for calculating details of the steel required to resist the given bending moment and shear force, applied to a given section. For theory see Chapter III.

Input Data

Section Width \( (W) \)
Section Depth \( (D) \)
Bending Moment \( (M) \)
Shear Force \( (V) \)
Option \( \) (compression steel is/is not allowed)

Output Data

Pass or Fail verdict
Minimum required area of tension steel
Number of bars of tension steel
Diameter of bars of tension steel
Spacing between bars of tension steel
Minimum required area of compression steel
Number of bars of compression steel
Diameter of bars of compression steel
Spacing between bars of compression steel
Minimum area of shear steel
Diameter of shear steel bars
Spacing of shear steel stirrups along the section
START

READ the input data

Set up a list of diameters of steel bars (British Standard diameters) in order of increasing magnitude.

Set the diameter of tension steel bars to the first in the list.

A

Set the diameter of stirrups (for shear) to the first in the list.

B

From the stirrup diameter, CALCULATE the cover to the tension steel.

2
CALCULATE the area ($A$) of tension steel required to resist moment ($M$).

Did the quadratic equation yield an imaginary root?  

NO:
Clear the bond failure indicator.  
(local bond)

YES:
The section is far too small. Set FAIL to 'IMAG' type

exit

CALCULATE the number of bars ($N$) whose area is greater than or equal to $A$ (which was calculated above).
CALCULATE the space between this number (n) of bars, with reference to the section width (W).

Is shear steel allowed?

YES

CALCULATE the number of stirrups that fit into the section consistent with the tension steel bars. CALCULATE the spacing of the stirrups along the member.

NO

Are the stirrups too close?

YES

NO

Have all bar diameters been tried? (If not, try next)

YES

NO
Are the tension bars too close?

Yes

Is the bond-failure indicator set?

No

No

Have all diameters been tried for tension bars?

Yes

Set FAIL to 'BARS' type.

Local bond failure. Set FAIL to 'BOND' type.

exit

No

Try the next diameter.

exit

No

Yes

5

4

A

C
CALCULATE the local bond stress on the tension bars.

Is the local bond stress too great?

NO

The tension bars have passed their tests.

CALCULATE the concrete (compression) resistance moment (Mrc).

6

YES

Increase the number of bars of tension steel (B) by one.

Set bond fail indicator.

3
Is $M_r$ greater than applied moment ($M$) ?

YES

PASS verdict to be given.

NO

Steel is needed to aid concrete.

Is compression steel allowed ?

YES

NO

Set FAIL to 'COMP' (compression fail)

exit
CALCULATE the area of steel required and clear the bond-fail indicator.

Set the diameter of compression steel bars to the first in the list.

CALCULATE the number of bars of the selected diameter, whose area is greater than or equal to the required area of compression steel.

CALCULATE the space between the bars of compression steel.

Are the bars of compression steel too close?

YES → 9

NO → 8
CALCULATE the local bond stress on the compression steel.

Is the local bond stress on the compression steel too great?

YES

PASS verdict to be given

NO

Increase the number of bars of compression steel by one.

exit

E
Is the bond-fail indicator set?

If yes, proceed to set FAIL to 'BOND' and exit.

If no, check if all diameters have been tried for the compression steel.

If yes, try the next diameter.

If no, set FAIL to 'BARS' and exit.