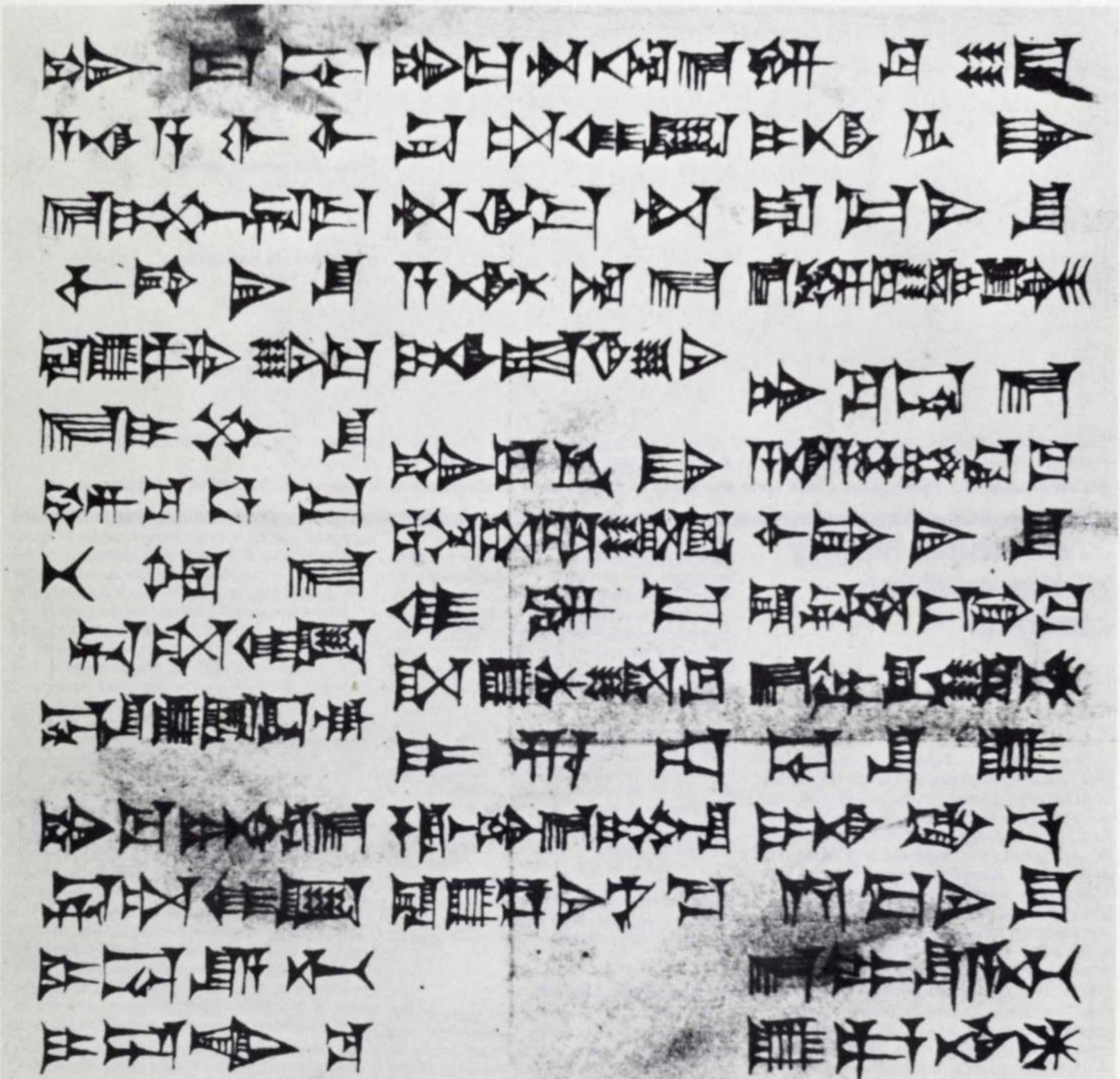
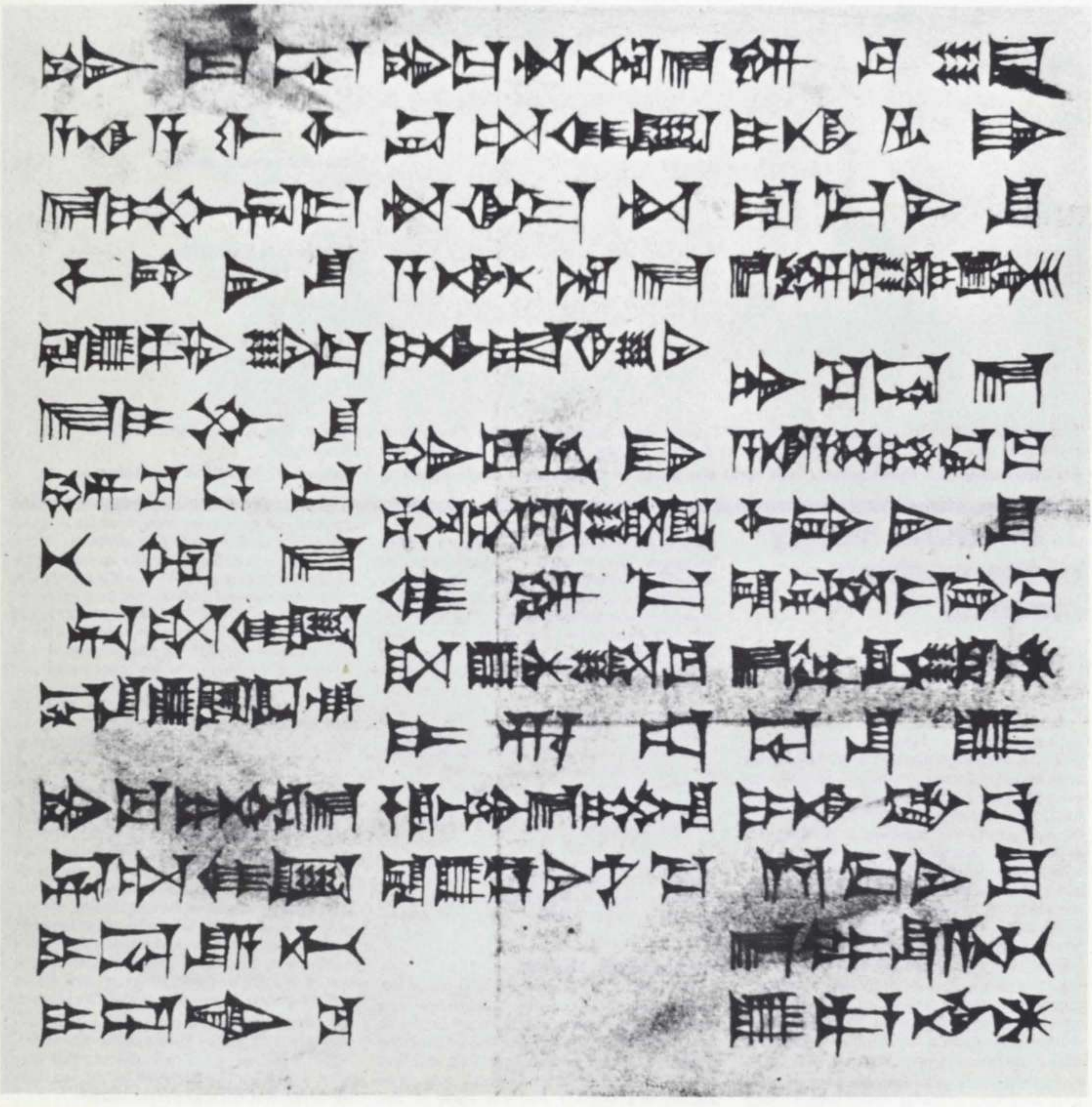
THE ARUP JOURNAL

JANUARY 1968



THE ARUP JOURNAL

JANUARY 1968



THE ARUP JOURNAL

Vol. 3 No. 1 January 1968
Published by
Ove Arup & Partners Consulting Engineers
Arup Associates Architects and Engineers
13 Fitzroy Street, London, W.1

Editor: Rosemary Devine
Art Editor: Desmond Wyeth MSIA

Contents

This is volume 3 of *The Arup Journal*. The 1967 issues were numbered volume 1 instead of volume 2. The *Journal* started in 1966.
Front and back covers: Photograph supplied by K. J. C. Clayden on return from his first visit to Kuwait.

Editorial note: All these papers will form the basis of a Technical Staff Meeting in February. Everyone is welcome.

The natural history of the washing machine

Bill Smyth

Let us consider the history of washing machines. The most primitive sort was a kind of tub with a paddle operated by hand. This was later worked by an electric motor but the operator had to turn the motor on and off, turn taps to fill and empty the machine and wring the clothes in a separate mangle. Then a time switch was fitted to the motor. The semi-automatic machine came later and could carry out washing, rinsing and spinning operations without the clothes being removed but the operator still had to fill and empty the machine and start each individual operation.

Now we have automatic machines in which a complete cycle of operations is carried out without human intervention. These machines work to a program, a sequence of instructions telling the machine what to do at each stage.

A slide rule is a simple analogue computer with an energy input of human elbow grease which can carry out one operation at a time. If we want to add the results of a number of multiplications (as in matrix multiplication, or weighting bending schedules) we have to write down the result of each one and then add them up as a separate operation. The mechanical calculating machine takes us a stage further, because it can add up the results of a series of multiplications without our having to intervene.

The slide rule corresponds to the primitive washing machine, the calculating machine to the semi-automatic and the computer to the fully automatic machine. The computer is a machine for doing sums which can carry out long sequences of arithmetical operations untouched by human hand. Just as the automatic washing machine carries out a particular cycle of operations, but you feed it with the clothes you want washed, so a computer program specifies a set of operations which is carried out on the numbers (data) provided by you.

The Computer Committee

At this stage I had better put my cards on the table. When the Computer Committee was first formed we saw one of our main problems as controlling the indiscriminate use of the computer. This is still so (and you will find echoes of it in Charles Wymer's and Poul Beckmann's articles) but this issue of the *Journal* has an ulterior motive which is quite the opposite. If there is still anybody in the firm who is frightened by the computer we would like to persuade him that it is really quite a domesticated creature and can be surprisingly useful not just for complicated problems, but to reduce some of the drudgery which is our lot, and leave more time for the important and interesting things. If, on the other hand, you are one of the cognoscenti, don't throw the *Journal* into the wastepaper basket. You will almost certainly find something in John Blanchard's article about the available programs and ways of using them which you didn't know. There is also an interesting contribution from Povl Ahm which reveals the horrid truth of why we have a computer. Charles Wymer's article tells the tyro how to go about using the machine. Alan Baker describes the actual set-up in the computer room, and Poul Beckmann writes about the use of the computer in the near future. Bill Hill gives a good example of the advantages of collaboration within the consulting engineer's computer group. Keith Law writes about the use of the *Elliott* Road Program, and David Taffs describes the frustrations of a troglodyte.

We intend to follow this up with a Technical Staff Meeting on the *Future of our Computer*, when it will be possible to discuss not only what programs and facilities will be available but what you think should be available.

Arups and the computer

Povl Ahm

In the beginning the computer was simply a scientific tool but very quickly it found its way into business in the form of data processing and into industry in the form of electronic controls and automation. It is probably also here that its most important future lies, taking away from us all our tedious routine jobs and leaving us free to think about what really matters.

Design

But the computer cannot design, surely ! ! Nor could we before we were taught how.

It may take some time and effort to teach a computer, especially since we ourselves often do not consciously—or even unconsciously—know the criteria we use for design. How do we, for instance, decide whether to use steel or concrete ? Once we have established these criteria there is no theoretical reason why we should not be able to convey them to the computer. But in practice we have to have one that is large enough and it has to be worthwhile. And this, as anyone who has just sniffed at the use of a light pencil (Cathode Ray Tube Display) will realise, is very much the question for the future.

Adding 2 and 2 together the computer does much better and faster than we do, because our brains are not designed to work that way, but used properly for conceiving ideas and forming intuitions, based on stored impressions and logical thinking, our brains are still infinitely superior to the finest computer that has ever been developed.

First steps

Our industry was rather late in realising the potential of computers. We ourselves started in a small way towards the middle 50's getting our linear equations solved by computers, but it was the work on the Sydney Opera House that really got us going.

We saw clearly the enormous possibilities in using the computer, in fact we realised probably earlier than most people in our field that certain problems simply could not be solved satisfactorily without it. But at the same time we were aware of the dangers in using the computer indiscriminately, mainly by asking it to solve the wrong problem or making the problem so complex that the answer would be meaningless.

The machine

The possibilities obviously far outweighed the queries and the partners quickly came to the conclusion that we had to be in from the start. After some years of service-centre experience it was decided that we should have our own machine and it was installed in September 1964. It was not actually ours. We rented it on very favourable conditions and it was for our sole use. It was an Elliott 803B, which at that time was classed as a medium-sized machine, and it would satisfy our demands for some years to come. So we thought !

We selected an Elliott machine partly because it seemed to be better value than its better known competitors from IBM and ICT, partly because Elliotts at that time were interested in the market offered by our industry and thus were prepared to accept a certain financial risk. I think they must have been as relieved as anyone when the arrangement did not result in a loss to them.

We never actually expected the computer to be profitable, not immediately anyhow, and a very simple calculation (*Newsletter* April 1964) showed this quite clearly. We expected to learn a lot at not too great an expense and to be able to tackle problems that had previously been out of reach. But first and foremost we hoped that most members of the firm would become familiar with the use of computers and use ours in the same way as their slide rules.

Snags

It did not go quite as we expected. As far as complex problems are concerned it went considerably better than expected and we quickly outgrew the 803B. It went so well in fact that within two years we had it replaced by the much larger (still only medium-sized) and faster Elliott 4120 (installed September 1966). This time we decided to buy it so that we could use it as many hours as we liked at no extra cost—but, admittedly, at a much higher initial cost. That has not made the balance sheet any more favourable, especially when we consider the greatly increased staff we need to operate the computer properly.

But as far as training and making people familiar with and interested in the computer is concerned it has not gone quite as well as we had hoped.

This is partly due to the difficulty of getting adequate programs developed and this problem was not made any easier by the change of computer, but partly it is due also to insufficient interest shown by us, the users. I hope this issue of the *Journal* and the follow-up planned by the Computer Committee, will go some way towards obliterating this fact.

Co-operation

We have for a long time realised the necessity for collaboration in our field on the development and use of programs because some of these programs require a very large effort and also because it is desirable that different users make use of the same programs in order to facilitate checking and communication. With this in mind we took the initiative in forming a group of consulting engineers in our own field—the Consulting Engineers' Computer Group—including G. Maunsell & Partners, Sir Alexander Gibb & Partners, Freeman Fox and Partners—and this group is now making some progress. We have also joined the National Computing Centre and PTRC* (Planning & Transport Research & Computation Co. Ltd.). These, of course, are much larger groups with more diversified memberships and are less likely to show an immediate return. In addition we are collaborating with the County Surveyors Society and the Ministries of Transport and Public Building & Works, especially on road and bridge programs.

The computer committee

The Computer Committee (P. Ahm, W. Smyth, A. Baker, P. Beckmann, J. Blanchard and C. Wymer) is set up to initiate and supervise all this work within the firm, helped by the permanent computer staff headed by Alan Baker and the Research and Development Group together with certain ad hoc working parties on specific problems. It is at the moment concentrating its efforts on getting a number of large general programs perfected (debugged is the jargon) and then it will be able to turn its attention to more ambitious programs for automation of certain of our activities and programs for 'finite element' methods. In this field the use of models as analogue computers linked to conventional digital computers seems to me to be very promising.

***Footnote by Bill Smyth**

The National Computing Centre is a government-sponsored body set up to co-ordinate and improve the use of computing systems, and, as Povl says, is not likely to show an immediate return.

I don't agree with him at all about PTRC which seems to me to be a lively and useful body. It is a non-profit-making company limited by guarantee which organises courses and seminars and will carry out computer work. Recent seminars included one on digital terrain models at which Keith Law talked about the Elliott Road Programs, and on horizontal alignment (of roads), and on urban planning and on transportation planning. They are also running short courses in various computer and programming subjects such as computer graphics.

Our present computer system

Alan Baker

Hardware

The basic components of the Elliott 4120 system are shown in Fig. 1 and specified in table 1. The interrelation of the units is shown in Fig. 2. The power-mad controller of the entire computer is known as the central processor. This actually consists of a control unit that orders everything, an arithmetic unit and a storage area. The storage area is variously referred to as the main store, the fast access store or the bit in the middle that is always proving too small.

The reserve or backing store consists of four magnetic tape units that sometimes think they are bottomless pits. However, they are catching on to the idea that they are supposed to file things and not lose them. The input is by mean of paper tape, eight tracks of holes in a 1 in. broad tape passed over a bank of photoelectric cells at a speed that is occasionally too fast for them.

The paper tape is produced by a machine called an off line printer with a typewriter keyboard. As the operator types the instructions to the computer the machine turns them into the code of holes on the tape. Output is in three forms. The first is paper tape which is slow and not used for the production of results. The main job of the paper tape punch is to make copies of programs, editing large quantities of irrelevant coding that got in by accident. If we wanted to make use of the paper tape output we could get the output typed out by the off line printer which can be used in reverse. The second and normal method of obtaining results is the line printer which is a kind of printing machine directly linked to the computer. The third means of output is a drawing machine called the digital plotter which draws by means of rocking paper backwards and forwards over a roller and traversing its pen head from side to side. That's all there is to it. One way in, three ways out, having skirted four bottomless pits.

Software

This is a general name applied to all programs presumably to denote that they are easier to change than the electronic or electromechanical parts of a computer. It is, however, suspected that it would be easier to use a soldering iron on occasions to achieve the desired results rather than attempt to change certain programs.

Software is supposed to come in two parts—the systems software which involves such things as the operating system and compilers, and applications software.

A compiler is a kind of translating instruction. The information fed into the machine is normally in a language such as *Algol* or *Fortran* which is not too difficult for human beings to learn, and you can regard the compiler as being a program which translates this language into the computer's own language (which is very difficult for human beings to understand.)

Software actually does come in two parts—later than the hardware and in a confusing number of issues. We are currently in possession of the thirteenth issue of the *Algol* Compiler and eagerly await the next. The *Fortran* Compiler is a young system in its second issue and therefore not yet blooded. There is also a machine language called *Neat* which exists in basic, elementary and advanced form. The other aspect of software, namely applications programs, is the only relevant part of the entire system. These are the programs that you actually use to answer questions. We have some of these programs. We will have more.

Who's who in the computer room

Alan Baker : *engineer*
John Jones : *mathematician*
Chandru Hira : *engineer*
Chris Doubell : *engineer*
David Taffs : *engineer*
Elizabeth Pitts : *programmer*
Cathy Sands : *secretary*
Peta Davis : *computer operator*
Sonia Wilson : *punch operator*
Ruth Goldberg : *punch operator*

SPECIFICATION FOR 4120 COMPUTER SYSTEM

Central Processor

24,576 words core store.
24 bit word.
6 microsec cycle time.

Magnetic Tape

Four
Data transfer speed 33,000 characters/sec.

Paper Tape Readers

Two
Input speed 1,000 characters/sec.

Paper Tape Punch

One
Output speed 110 characters/sec.

Line Printer

One
Printing width 120 characters
Printing speed 300 lines/minute

Digital Plotter

One
Drawing width 12 in.
Drawing speed 1.5 in. per sec.
Accuracy 1/200 in.

Control Teleprinter

One

Central Processor

Arithmetic Speed	Fixed Point	Floating Point
Addition/subtract	12 microsec	270 microsec
Multiplication	65 microsec	550 microsec
Division	66 microsec	560 microsec

Right: Figs. 3 and 4 show two general views of the computer room

All photographs by Colin Westwood

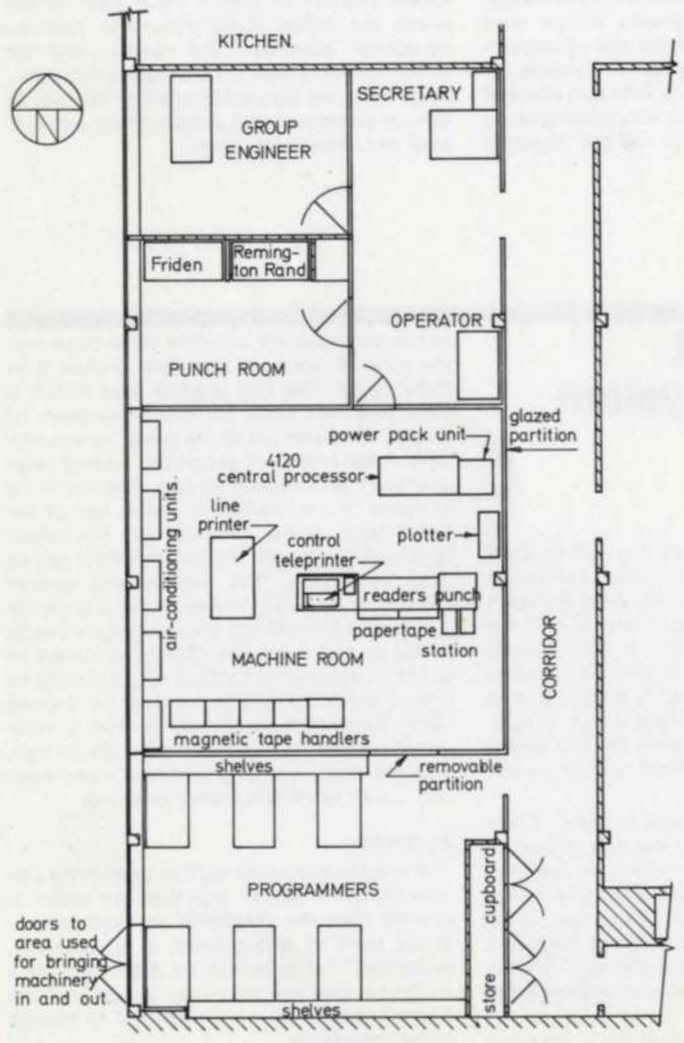


Fig. 1 Computer group layout

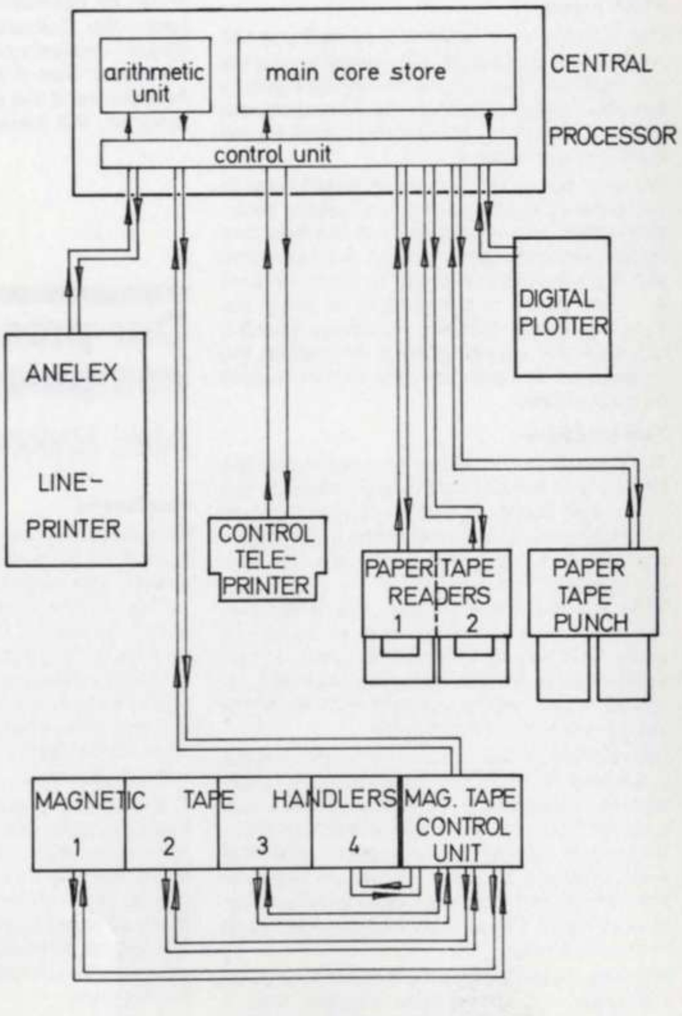


Fig. 2 Block diagram of computer system

Fig. 5 Control teleprinter

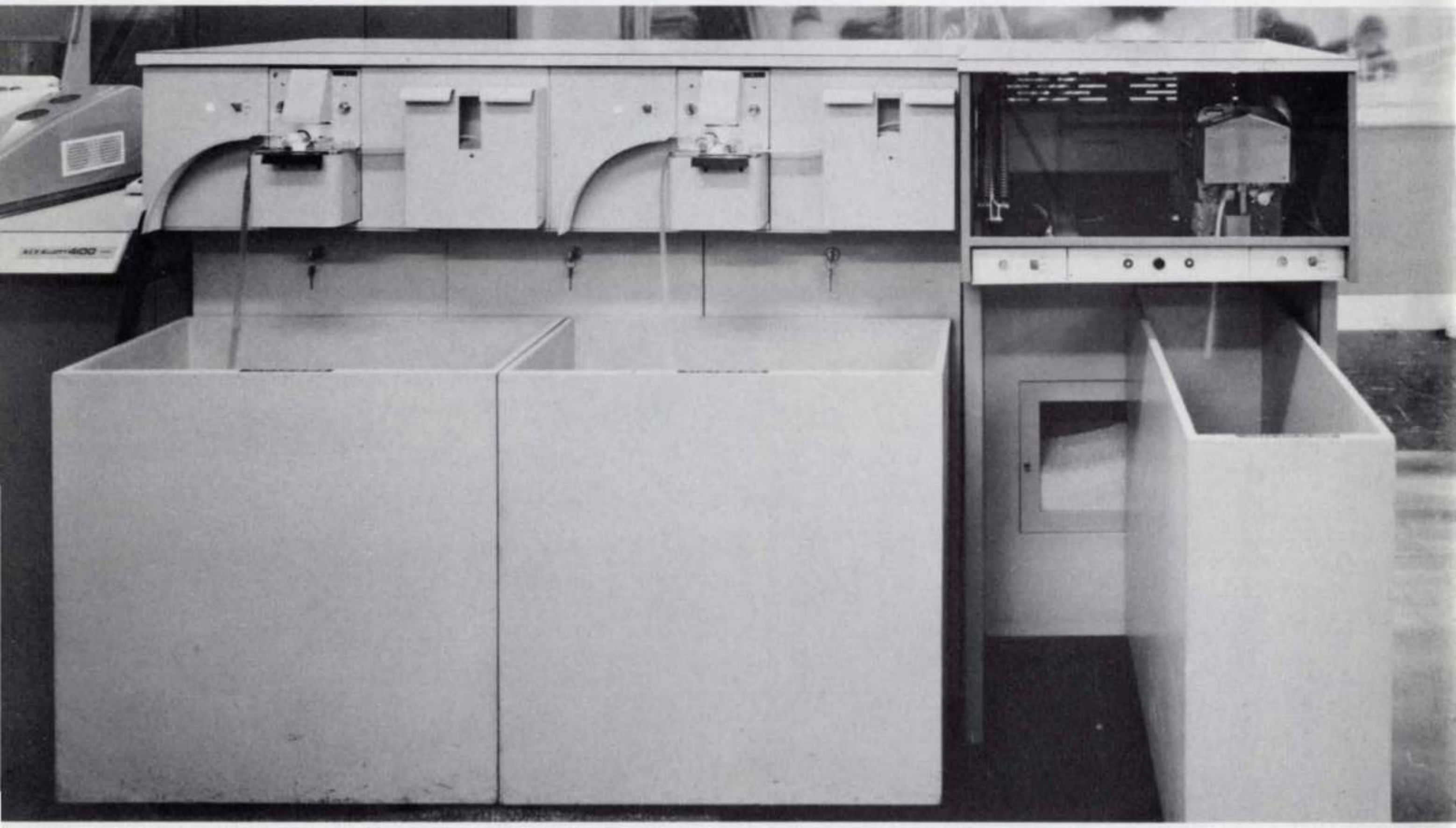
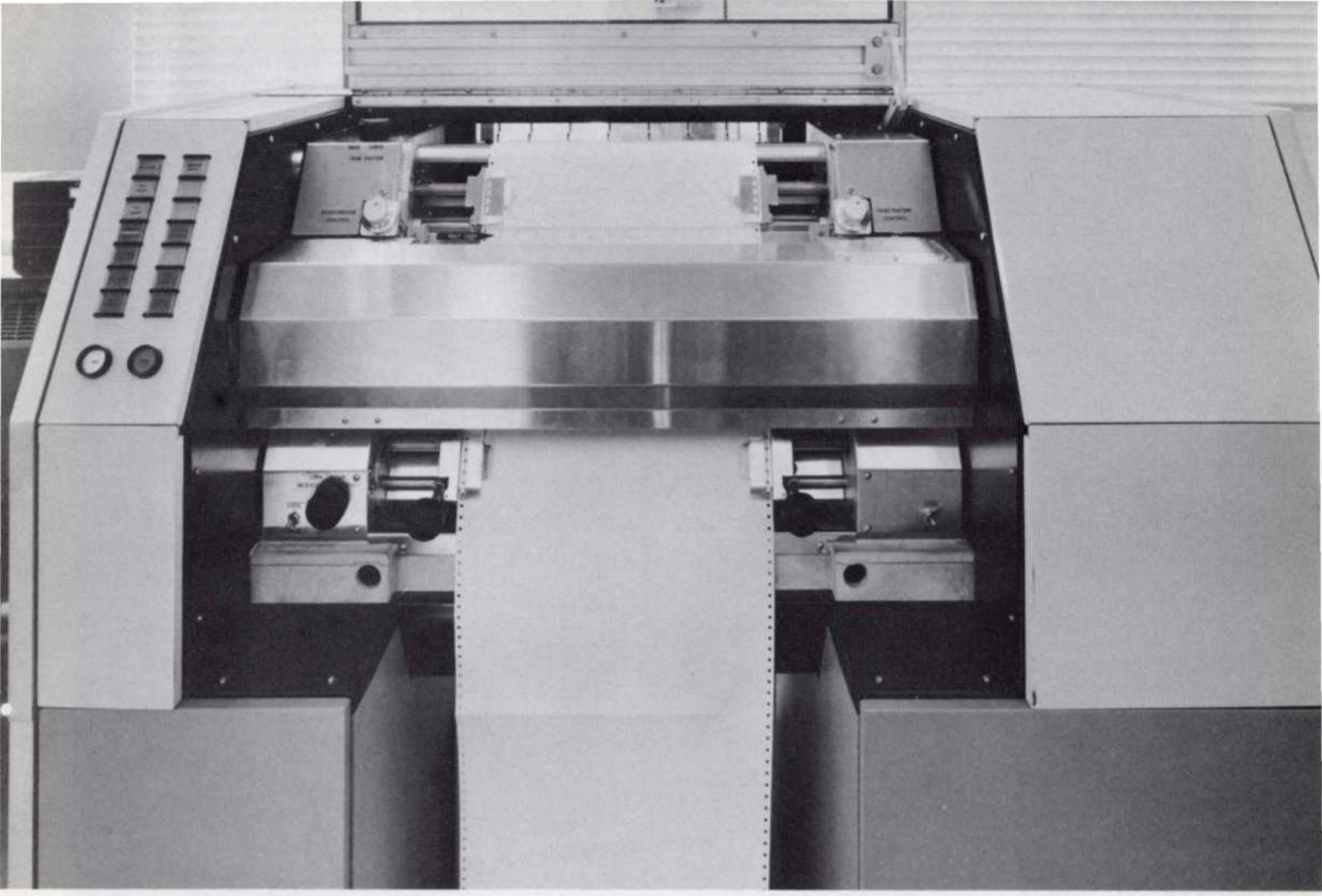
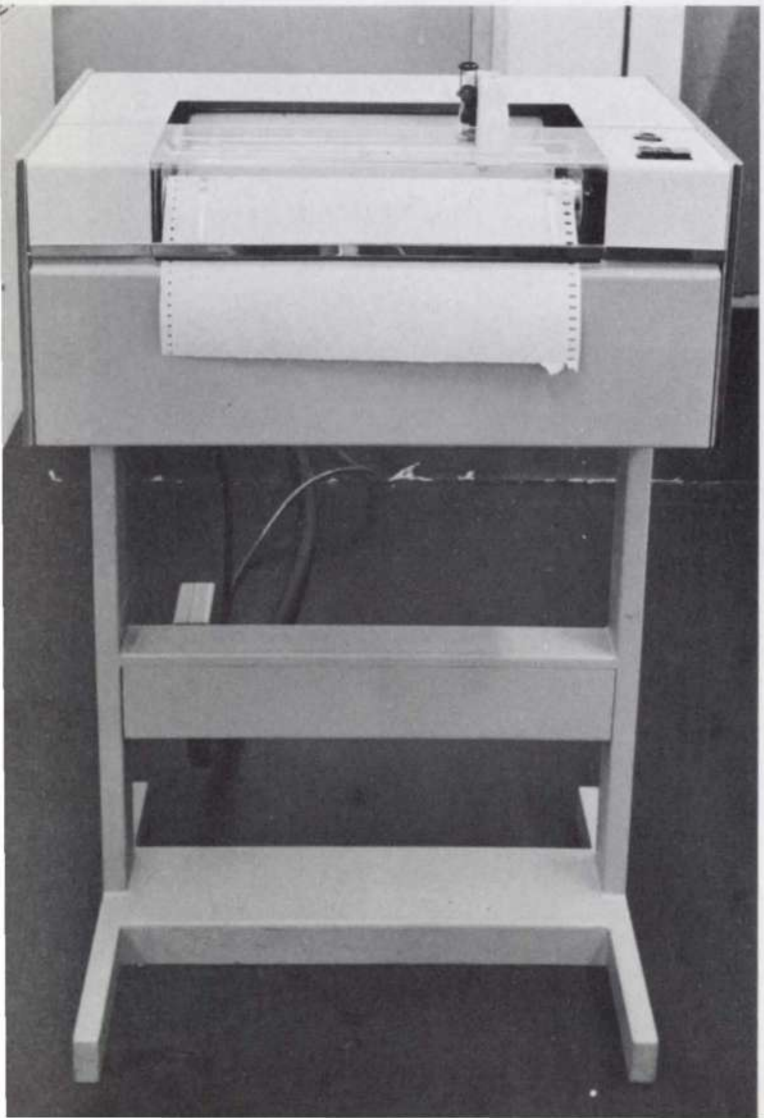


Fig. 6 File 13 or paper tape station

Fig. 7 Magnetic output units

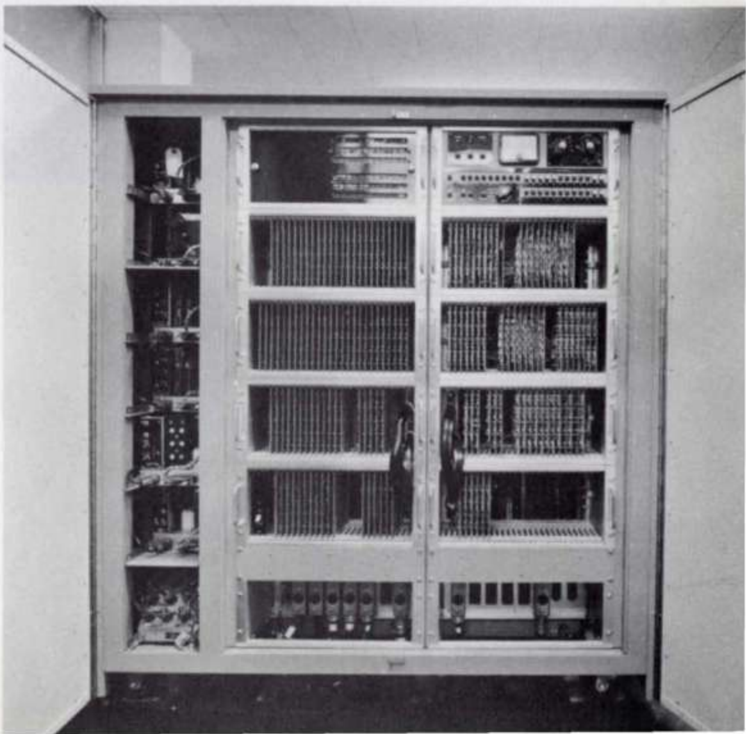


Above: Fig. 8 Line printer



Left: Fig 9 Digital plotter

Below: Fig. 10 Central processor.



The available programs

John Blanchard

Specifications for the available programs for our computer are, or should be, kept on each floor. This article describes these programs in a different, if not a more readable, way.

This article, written in September, has to predict the situation four months hence. A computer generation lasts about four years so that the task is analogous to predicting, before a child is born, the age at which he will first make a sound that can reasonably be construed as a word. It could be worse, at least the child has been conceived, nevertheless there may be inaccuracies in the descriptions.

Structural framework analysis

These programs easily head the popularity poll. They have, more than all other programs put together, relieved engineers from the drudgery of calculations; freeing them for the task of preparing more data for yet more complicated structural analyses.

They will analyse two or three-dimensional frameworks by linear elastic methods, assuming that members between joints are thin, straight and uniform with specified constant section properties. The program first prints out the lengths and directions of all members, calculated from the given joint co-ordinates. Then they produce, for each load-case, the movements of each joint and the forces and moments acting at each end of all members. The movements are given with reference to the main co-ordinate axes; forces and moments with respect to co-ordinate systems local to each member. Their units will be consistent with those of the input loading, dimensions and section properties.

The *Space Framework Program* is used for the analysis of three-dimensional problems. Two-dimensional structures are treated as *Plane Frameworks* if all forces and movements occur in the plane (e.g. a wall carrying vertical loading) and as *Grid Frameworks* if all the forces and moments are at right angles to the plane of the structure (e.g. simple floor slab).

Loading

The applied loading may include uniform load along any member, point loads anywhere on a member or forces and moments applied at a joint. If required, self weight can be included automatically. Loads may act at any angle but must be specified as components in the directions of the main axes, uniform loads in terms of a unit length which is parallel to a main axis. Complex loadings and temperature effects on members are dealt with by applying the corresponding fixed-end moments and forces as loads on the joint at each end. The computed moments and forces at the ends of the member must be corrected finally by subtracting these fixed-end actions.

Additional load-cases can be included with little extra cost. So when in doubt whether a particular load-case is needed, it is probably better to include it. If you do not and later find that you want this load condition, it will cost a lot more to re-run the program.

There is no way in which changes of geometry or member properties can be introduced without a complete and costly re-run. If possible, therefore, the programs should be regarded as strictly analytical tools to check and refine an approximate design already made. This design will also serve as a rough check on the computer results, being particularly useful for the detection of errors in the input data.

A rather subtle error sometimes occurs when an analysis is being repeated with a modified structure simulating some condition during construction. It may be that in the modification members have been omitted leaving a part of the structure (perhaps an isolated joint) which is unstable. Even if this part of the structure is unloaded the machine is likely to overflow and stop. What has happened is that a very small stray load has found its way to the joint and produced an infinite movement.

Springs

A facility is provided for supporting joints on springs (extensional or rotational). This can be used to represent elastic foundation conditions and also to investigate local effects on a large structure. Parts of the structure sufficiently remote from the loaded area can be removed and replaced by springs of equivalent stiffness determined approximately or perhaps by a separate computer analysis. In difficult cases where several springs are used to simulate a part of the structure, some of the spring stiffnesses may be negative. This is acceptable to the program provided the negative stiffnesses are not too great. Similarly a part of the structure can sometimes be simulated by a single member whose equivalent I needs to be negative. This, again, may be acceptable.

Another use of springs is for the application of a pattern of forced displacements to a structure. Extremely stiff springs are placed at the joints which are then loaded so as to produce the required extensions of the springs. If the springs are stiff enough, these loads are independent of the stiffness of the structure and can be easily calculated.

Semi-rigid joints

Joints between members are normally assumed as absolutely rigid and monolithic. Earlier programs allowed for the introduction of pins at the ends of members but this facility has been replaced by the more general one of semi-rigid joints. With this, a coiled spring is introduced between the end of a member and the joint. The rotational stiffness of this spring is taken equal to the member stiffness divided by a coefficient specified in the data. If the coefficient is zero then the joint is completely monolithic; if it is infinity (in practice 50 is infinite enough) the connection is pinned. For intermediate values, semi-rigid joints, such as cleated steel connections, can be simulated. This facility is available for all bending moments in plane, grid and space frameworks but not, at present, for torsion moments.

This device could be used to predict collapse mechanisms for frameworks or slabs by an iterative procedure (perhaps, in the future, automatic) by which the spring stiffnesses are given values depending on the bending moment computed in the previous analysis.

A version of the *Space Framework Program* is available in which all members are assumed pin-ended.

The stiffness method

All these programs use the *Stiffness Method* of analysis. This is a generalised slope-deflection method in which the forces and moments acting on the end of a member are expressed in terms of the unknown displacements and rotations at both ends of that member (for plane and grid-frames three forces or moments in terms of three rotations or displacements, for space-frames six of each). Since three equations of equilibrium (six for space-frames) can be written down at each joint we have 3j (or 6j) equations in the 3j (6j) unknown movements, where 'j' is the number of joints in the structure. These equations are solved to give the movements from which the forces and moments on the members can be found. The collection of coefficients in the 3j equations is known as the stiffness matrix of the structure because it is analogous to the stiffness of a single member.

Band width

Thus a grid framework with 500 joints would require the solution of 1,500 simultaneous equations, a formidable task even for a computer. Fortunately, most of the coefficients in the equations are zero; for example, if joint 70 is connected only to joints 40, 69, and 71 then the only non-zero coefficients in the 208th, 209th and 210th rows would be those in positions 118, 119, 120, 205 to 213. It takes almost as long to locate and multiply by zero as it does by a finite number, so that the presence of all these zeros is of no help unless we use a banded stiffness matrix. A diagonal band of coefficients is chosen to include all the non-zero ones; all the coefficients outside this band can be forgotten and will not waste valuable storage space.

The band-width of a structure is defined as the greatest difference between joint numbers at the ends of any member (although the actual band width of the stiffness matrix is more than three times this). The way in which the joints are numbered is therefore important and should be chosen to make the band-width as small as possible; so that the solution of the equations is as quick and cheap as possible. Indeed if the band-width is too large the problem may be beyond the capacity of our computer.

Minimizing the band-width

There is no known general method for finding the best numbering system or of determining the smallest possible band-width. Anyone who could find such a method would make himself very popular. He might even have a branch of applied mathematics named after him.

The band-widths of the structures in Fig. 1 are more or less obvious, but it is not clear that the plane framework shown in Fig. 2 (one-quarter of a stiffening diaphragm for a 70 ft. diameter, 600 ft. high chimney at Didcot Power Station) with a band-width of 6 was numbered in the best possible way.

The problem is made more complex by the possibility shown in the plane framework of Fig. 3 that the addition of extra joints may reduce the band-width. Joints 21, 36, 51 and 66 were not needed structurally but their introduction reduced the band-width from 7 or 8 to 5. Without them the structure would not fit into the program then available.

In Fig. 3 member 6-71 represents a rib under construction at the Sydney Opera House. This rib is supported by needles such as 10-5, 14-19, etc. spanning from a steel erection arch represented by member 1-74 and by the existing rib assumed immovable at points 5, 19 etc. Since the ribs and the steel arch lie in different non-parallel planes this shows that a three-dimensional problem can occasionally be reduced to a plane problem.

Sizes and costs of framework analyses

A convenient unit to use when considering the allowable sizes of frameworks is the *srini*. Strictly, a structure has a size of one *srini* when the output for one load-case completely fills a standard box. This is roughly equivalent to a framework with 500 joints and 900 members (A $\frac{1}{2}$ *srini* structure would have 250 joints etc.)

The maximum size of plane or grid frameworks that can be analysed by our own computer is about 0.6 *srini* with a band-width of 15. A greater band-width is not allowable, with smaller band-widths larger structures would be accepted. More complex structures of up to 2 or 3 *srini* would have to be sent away to a larger machine at Boreham Wood.

For space frameworks the maximum size is about 0.3 *srini* with a maximum band-width of 9.

The cost of analysing a plane or grid framework of 0.35 *srini* with a band-width of 9 and one load-case is about £33. One of half this size would cost about £14.

Torsion in grid frameworks

A difficulty frequently met with in grid frameworks is that the computed torques are higher than can reasonably be dealt with. It is standard practice to prevent this by using reduced values (say 50%) for the torsional stiffnesses in the input data. The justification quoted is that, in fact, high torques do not occur, they

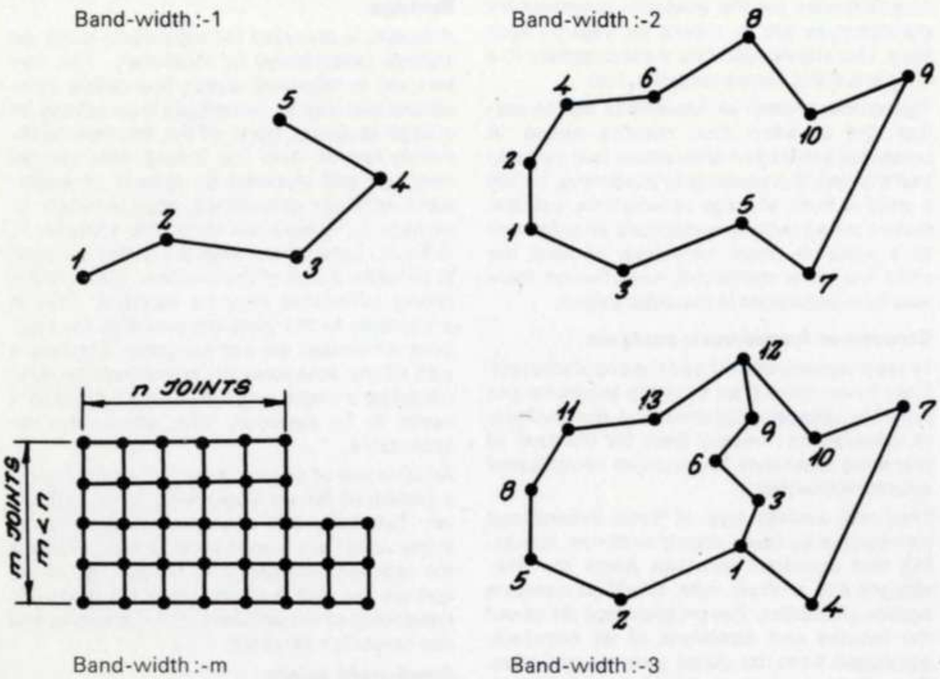


Fig. 1

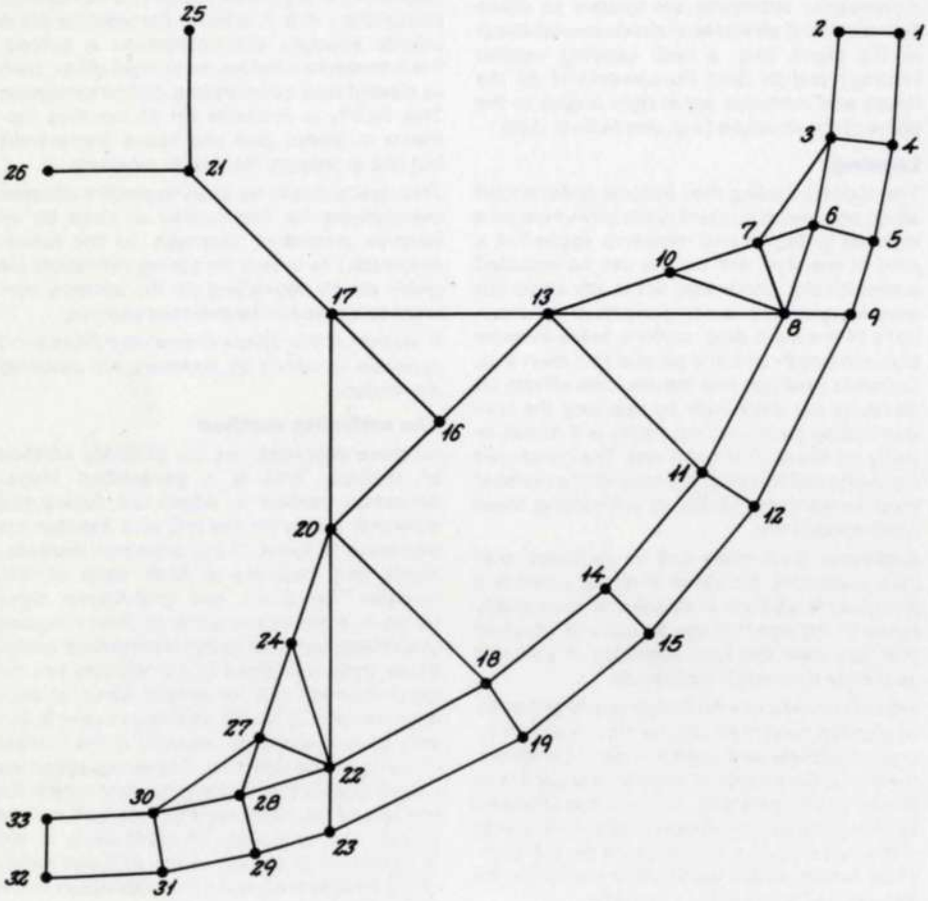


Fig. 2

have been reduced by some yield mechanism. It is true that some plasticity effects do occur in reinforced concrete members under torsion but it is thought to be limited and followed by a sudden failure. More information seems necessary before this practice becomes sanctified.

Solid slabs

It can be shown that the behaviour of a solid slab in bending can be accurately represented by a regular rectangular grid framework. It turns out, as might be expected, that the correct value of I to use is that of a width of slab equal to the spacing of the grid members. But, surprisingly, this value is also the correct one to use for the GJ/E of the member. This is true also for cellular slabs with top and bottom slabs, but coffered floors without a bottom slab are probably more accurately represented by calculating the I's and J's of tee-beams in the normal way. If a grid member represents several ribs the I's and J's would be the sum of those of the individual ribs. The correct treatment of the edge member in solid or cellular slabs is open to doubt, but good results have been obtained by placing this member on the boundary and giving it section properties of one half of those for the inner members.

High values will be found for the computed torques in solid slabs. These will in fact occur and are resisted in quite a different way from those occurring in beam grids. They will combine with the bending moments, in a way precisely similar to that in which shear stresses combine with direct stresses, to give, at some angle to the original axes, principal bending moments without torques. Very often, torques near mid-span can be neglected and those near supports will be catered for by the requirements of CP114 for corner steel, but spot checks should be made to confirm this.

Grid framework to solve heat transfer, torsion and permeability problems.

Since the deflections of a suitable grid framework satisfy the plate equation it follows that the moment-sum Mx+My at any point satisfies Laplace's equation. This equation governs many physical problems including the steady-state heat transfer problem discussed in **Technical Paper 4* by J. Melling and M. Johns.

Thus, the moment-sums throughout a grid with suitable edge conditions and unloaded (a load would imply a heat-sink or source) are proportional to the temperature distribution in a section of the same shape. If there is no boundary layer then the appropriate edge condition is simply-supported with an applied bending moment proportional to the boundary temperature. With a boundary layer, the support would lie outside the boundary and connected to the grid by a beam of suitable stiffness. In effect, the relaxation equations of *Technical Paper 4* are being solved without the labour of writing them out explicitly.

Similarly with a prescribed edge loading on a suitable grid the moment-sum could represent the torsion function and the shears in the members would be proportional to the torsional shear stresses in the simulated section. An additional step is required in this problem, for the shears would have to be integrated in some way to determine the applied torque.

Also the moment-sum could represent the potential function to solve the problem of flow of water through permeable strata. There is a variety of boundary conditions that could be reproduced on the grid but that of the undetermined free surface, which frequently occurs, would be difficult to handle satisfactorily.

Plane stress problems by the finite element program

Since the grid framework works so well for solving the problem of bending of a plate, it might be thought that the plane framework could be used to solve plane-stress problems, i.e. that of a wall or deep beam carrying loads in its own plane. Experience has shown however that for many cases it is not possible to devise a framework which will give sufficiently accurate results. Theoretically the best solution is obtained by a rather artificial equivalent grid framework but this is difficult to use and the allowable boundary conditions are restricted.

A *Finite Element Program* has therefore been written which will deal with this sort of problem. The wall is divided into not more than 10 x 10 rectangular pieces which are handled in much the same way as the members in a framework program. The program has not been completely tested but appears to give very accurate estimates of deflections. Some care is needed, however, in the interpretation of the stresses that are printed. It is proposed eventually to incorporate wall-like elements as allowable members in the framework programs.

Shell structures by space frameworks program

Since the principal forces in shells act along the surface it should follow from the last section that a shell roof could not be simulated properly by a space framework. However there are differences between the two problems; in a shell, the in-plane forces are not the only actions and they change more slowly than in a wall. Furthermore, a shell designer is usually satisfied with a less rigorous solution and is pretty thankful to get stresses which satisfy equilibrium. At least a space framework program will give him those.

It is probable therefore that an equivalent framework can be found to represent satisfactorily a given shell structure. Common sense seems to be the only guide as to what stiffnesses should be given to the equivalent members. Research into this question was started in the U.S.A. but has apparently been dropped in favour of the finite element approach. Eventually, no doubt, a finite element program for shells will be available to us. Until then, if a theoretical solution is not known, we can use either an equivalent space framework or Alistair Day's method of dynamic relaxation. This is a general method useful for many types of structural problem but standard programs are not available. A new program has to be written for each type of shell.

Grid frameworks with shear deformation

The framework programs allow for deformations due to direct thrust as well as bending and torsion moments, but neglect the deflections due to shear forces. A variant of the grid framework program makes allowance for shear deformation. This is not a purely academic exercise for it enables an important type of member to be represented.

Where bridge decks are cellular but without transverse stiffening diaphragms, lateral distribution of load is effected by only the top and bottom slabs. They can readily transfer a

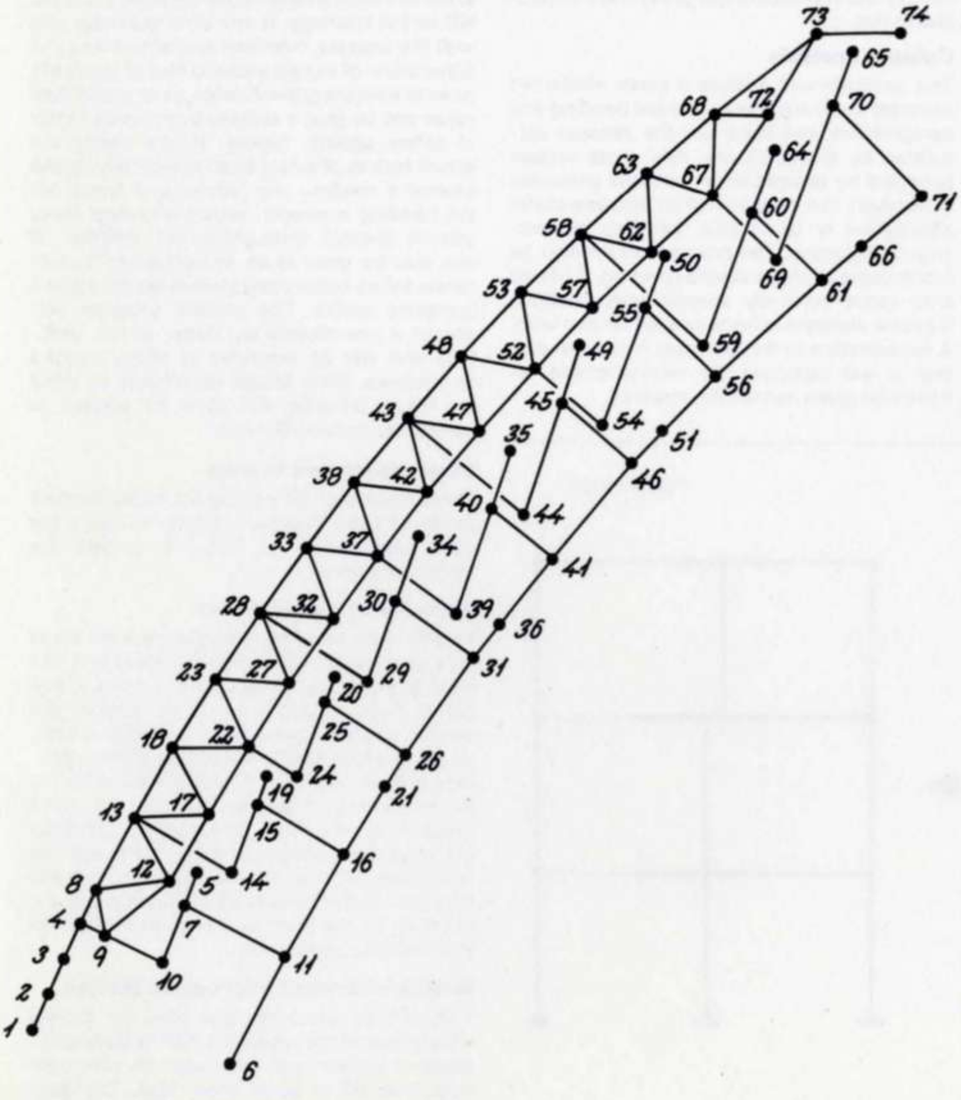


Fig. 3

*** Editorial Note** *Ove Arup & Partners Technical Paper 4—The determination of the temperature variation of partially exposed columns, by J. Melling and M. R. Johns. 1967.*

constant bending moment for this produces direct forces in the slabs but shears can only be transferred by bending in the relatively thin top and bottom slabs. An equivalent member has to be used therefore which is very weak in shear. When the soffit slab is sloping, as it may be near the edge of a bridge deck, two members in series have to be used but they give a rather rougher approximation to the true structure.

This variant would also be worth using for a grid of latticed beams for which shear deformations are often important. The automatic self-weight facility is not available with this program.

Infinitely stiff members for shear walls, folded plates and cellular boxes

In the framework analysis each member is represented by a thin straight line. Where an action applied at the surface of a member would produce a significantly different effect from one applied at the centre line, a rigid outrigger can be used to represent the condition 'plane sections remain plane'. The rigid outrigger is a member given section properties so large that its movements are negligible. It can be used whenever you have a plate whose proportions are such that the ordinary beam assumptions are valid. It is useful, for instance, where a shear wall forms part of a complete framework as in Fig. 4. Here it would be sensible to give the outriggers the section properties of a storey-height of wall.

The same device can be used in three-dimensional structures, particularly to represent folded plates provided that each plate can be satisfactorily approximated by beam theory and provided that a structural model with connections between plates at discrete points rather than a continuous connection is acceptable. It obviously is when there are enough connections. Thus, a folded plate structure or a cellular box (which is a sort of folded plate) can be represented by an equivalent space framework. The resulting structural model obviously has its drawbacks but they are probably no greater than those involved in the usual methods of analysis using Fourier series for which, anyway, there are no computer programs available to us.

Reinforced concrete element, analysis and design

Apart from the above programs which deal with the analysis of the structural frame as a whole, six programs have been written to analyse or design isolated elements of a reinforced concrete structure.

The first two are the most useful in that they do a lot of work for a small amount of data preparation. The last four perform rather elementary operations and are not (especially the column loading program) usually worth the effort and time involved in getting the program and data tape into the machine; although they may prove economical where many elements are being designed together.

The main reason for their existence is that they are expected to form parts, eventually, of a bumper program designing a complete structure. It is visualised that, for such a program, a comparatively small amount of data would be required, describing the basic characteristics and dimensions of the building. This could be stored permanently on magnetic tape and amended as the architect's and other changes are made. This is a long way in the future but meanwhile some of the component parts of the program are undergoing a probationary testing period.

Continuous beams

This program will calculate bending moments and shears (by normal elastic theory) for up to twelve continuous spans of given rectangular, L or T-beams, making the 15% adjustment allowed by CP114.

It then prints out the required areas of tension and compression steel and of shear stirrups at the supports, quarter-points of the span, and at point loads. Compression steel is limited to one layer. The original program does not treat the effect of incidental live loading rigorously so should be used with care when the live load is high compared with the dead load. A revised program will treat live loads exactly but will necessarily prove more expensive to run.

Column analysis

This program will analyse a given reinforced concrete section under combined bending and compression and print out the stresses calculated by elastic theory. Any cross-section bounded by straight lines (virtually unlimited in number) can be specified so that one useful application is to bi-axial bending of rectangular columns. The program might also be worth using to check simple bending of beams with those curiously shaped cross-sections that one sometimes finds oneself landed with. A modification to this program is proposed so that it will calculate the reinforcement required for given permissible stresses.

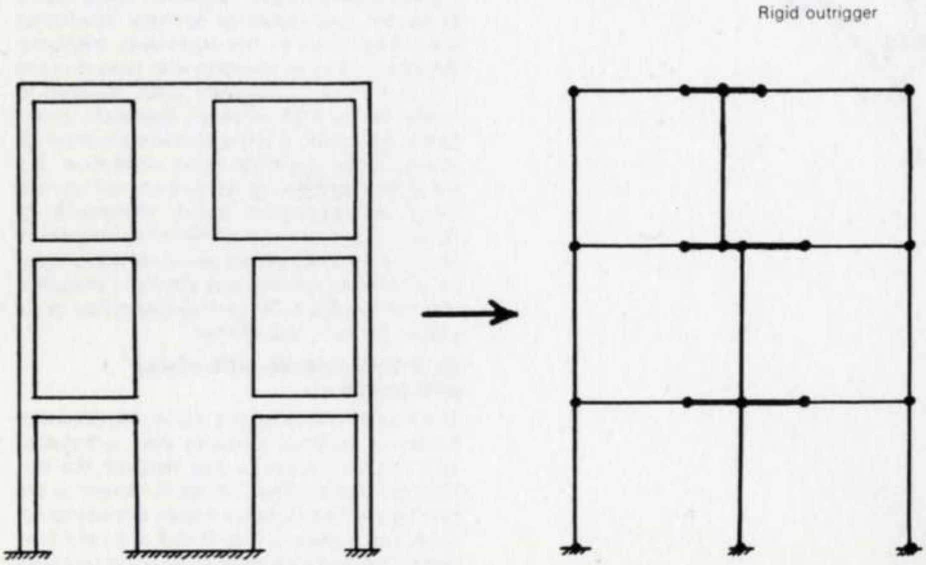


Fig. 4

Column loading

This program prints out column loads in each lift of a column, allowing for live load reduction factors and for increasing the loads to cater for bending effects in the way suggested by the Danish Code. The considerable input data required includes the finish, live load, thickness, length and breadth of all slabs carried by the column; the length, depth and breadth of all beams; the length and intensity of live loads such as partitions.

Column design

This gives the optimum column size and reinforcement to carry a specified axial load. Apart from allowable stresses, the minimum and maximum sizes and the preferred steel percentage must be fed in. If no suitable column of the allowable dimensions exists with this steel then the program investigates other percentages of reinforcement (above 0.4%), printing out an adverse comment if more than 4% is required.

Column footing (rectangular plan shape design)

This program finds the optimum length and width of the footing given: the axial load and the overturning moments in two directions, the column size, the upper and lower limits of the footing dimensions, and the allowable ground pressure. It also prints out enough information to determine the actual ground pressure distribution. This is not as trivial as it sounds when uplift occurs with bending in two directions. This program will soon be extended to investigate the thickness and reinforcement required.

Retaining wall

This will design a cantilever wall to retain Code of Practice earth surcharged by footpath HB or HA loadings. It will print out base and wall thicknesses, heel and toe widths and the dimensions of the downstand key (if required) so as to limit the ground pressure to a specified value and to give a specified minimum factor of safety against sliding. It also yields the actual factors of safety against overturning and against a shallow slip failure, and prints out the bending moments, shears and steel areas (elastic theory) throughout the member. It can also be used as an analytical tool to calculate safety factors and steel areas for a given complete profile. The present program will accept a pre-determined batter of the wall-face and will be extended to allow stepped thicknesses. With bridge abutments in mind the future program will cater for vertical or horizontal loads on the wall.

Steel masts and towers

These have been developed by Andre Bartak's group and the Computer Group to allow the rapid preparation of G.E.C.'s tenders for microwave lines.

Analysis of guyed masts

This program analyses the effect of wind loads on a guyed mast with pinned or fixed feet, the mast being a single member or a three-sided lattice truss. Apart from displacements the output includes forces and moments for the single member mast and stresses in the members of the lattice truss. A special version allows for the effect of torsion due to, say, wind pressure on eccentric bowl aerials provided the guys are suitably arranged. The analysis is complex because allowance has to be made for non-linear effects such as sag of the guys, buckling of the mast and eccentric loads on the mast due to sidesway.

Design of braced microwave towers

This will calculate member sizes for towers which may be triangular on plan with double-angle or tubular legs or square on plan with double-angle or equal angle legs. The geometry must be pre-determined, up to three lifts with differing leg slopes are permitted, and the number and type of bracing panels

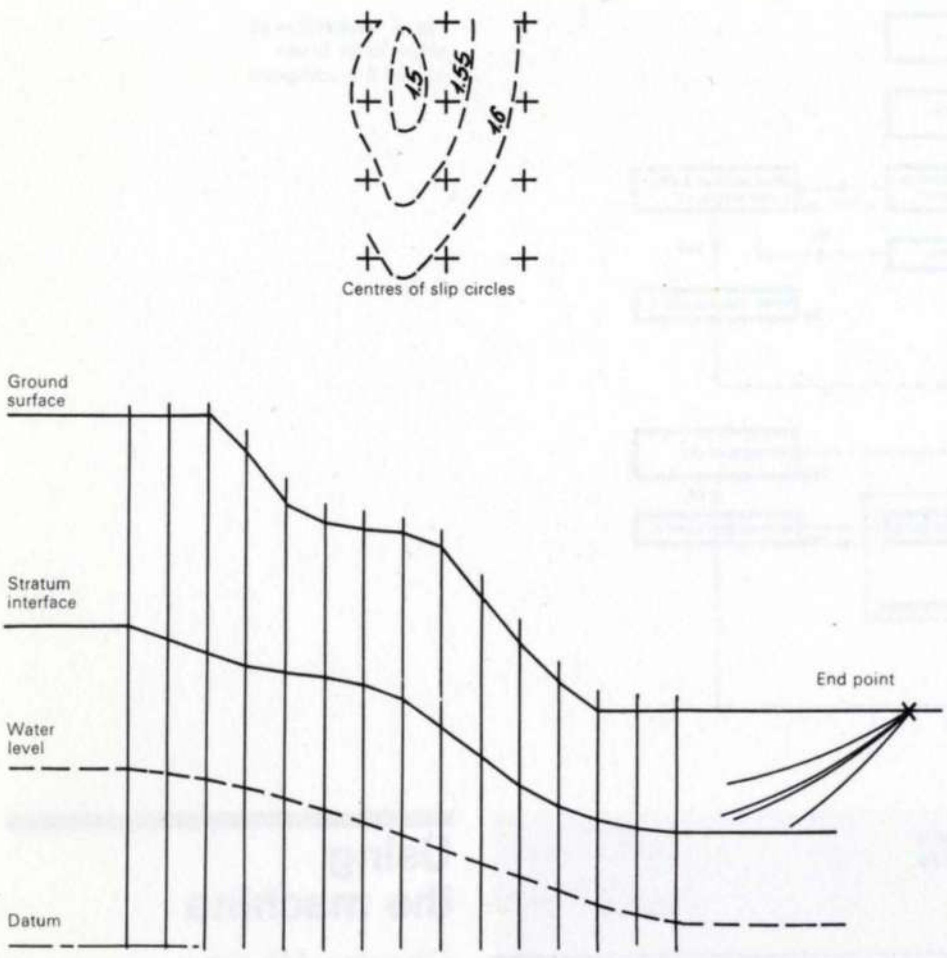


Fig. 5

must be quoted. Other input data required are the wind speed, positions and diameters of aerials, additional loads from wave-guides and a series of trial member sizes.

The program now calculates the wind loading, allowing for shadow effects, and determines the member forces by a pin-jointed analysis. A miniature section hand-book stored with the program is then consulted, the member-sizes adjusted accordingly and the whole process repeated until all the members are satisfactory from a load-carrying point of view. Now the rotation and twist at the top are calculated and the leg sizes increased as necessary to ensure that these movements are below those specified as necessary for efficient signal transmission.

Soil mechanics

These programs should be used under the benevolent eye of an engineer from the Soil Mechanics Group. This is not only because the selection of soil strengths contains traps for the unwary but also because where the programs call for some estimate of the position of the worst slip surface an experienced guess can save a lot of the machine's and the engineer's time.

Circular slips

This program will calculate factors of safety against failure of an earth slope as shown in Fig. 5 by circular slips with various centres and radii. The centres tested lie on a grid prescribed by the data and for each centre the radii change with given increments below a maximum radius which will pass through a specified end point. From the results a contour diagram of safety factors can be drawn in the grid of centres; thus one can find either the minimum safety factor or that one has chosen the grid in the wrong place. As a rough guide the safety factor contours are very often elongated vertically with the minimum lying approximately above the mid-point of the slope.

Two strata of different properties can be used and the effect of a water-table included (stability is calculated on effective stresses). The earth mass is divided into any sensible number of vertical slices of equal width. By specifying for each slice the heights above some datum of the ground surface, strata interface and water-table quite complex slopes can be represented. Not content with this, the program will, if required, repeat the calculations for water-tables lowered by specified uniform draw-downs, or for different strength properties.

A variant of this program is suitable for clays. It takes the angle of friction as zero but allows values of cohesion varying arbitrarily with depth.

Non-circular slip

In this program the factor of safety is calculated for an assumed slip surface which can be of any shape specified by its height on each slice. As for the circular slip, two strata and a varying water-table can be considered. The slice widths need not be constant and are best chosen so that the slip surface intersects the water-table and strata interface on the edge of a slice. This is even more a program for the expert, for who else would be rash enough to believe that he had chosen the worst surface.

Foundation settlement

This calculates influence lines for the settlement of the surface of a semi-infinite earth mass due to unit point load applied to the surface. It then uses these influence values to calculate the settlement under a building applying a given loading to the ground. Either elastic, incompressible theory or consolidation theory can be used. It should be noted that no account is taken of the stiffness of the raft or of the building itself.

Miscellaneous programs

Simultaneous equations

This will solve simultaneous linear equations in up to about 100 unknowns with any sensible number of right-hand sides. More unknowns might be acceptable but who is going to write down more than 10,000 coefficients to find out?

Properties of plane sections

For any given cross-section that can be split into rectangles, triangles, circles or complements of a quadrant of an ellipse (to reproduce fillets) this will print out the area, direction of principal axes and the principal moments of inertia and section moduli.

Concrete cubes

This will print out the mean and standard deviation of a given set of cube-test results after excluding those results outside a certain range (this removes the possibility of the date being included in error). The program also records the sums of cube strengths and their squares. These can later be added to any new test results thus avoiding the need for storing or re-listing the original results.

Perspective drawing

For anyone with the hardihood to calculate and list the co-ordinates of up to 1,000 points, this will draw on 12 in. wide paper a perspective drawing of an object from any required vantage point. Unfortunately, like all perspective programs, it is unable to allow for the fact that near parts of the object may obscure those parts further away. It is particularly good at drawing the Durham Footbridge.

Road design package

This collection of programs prepared by Elliott's will plot a road-alignment to pass through given points incorporating the necessary transition curves. Given details of the terrain and allowing for super-elevation, they will plot longitudinal and transverse cross-sections and make the calculations for cut-and-fill, etc.

An additional geometry package due to Arups calculates offsets from given chords of the road centre-line and of the inner and outer channels.

Pert

This program due to Elliott's performs a conventional network analysis for up to 4,000 activities and 3,000 events and produces the usual schedule of earliest and latest start dates, etc. This could well prove useful for the organisation of the design and details for a complex project if sufficient confidence were felt in the estimated times for the various operations. The program includes additional sorting facilities so that, for example, the latest dates of the various architect's details could be output separately in chronological order.

Elliott library programs

A catalogue in the computer room lists the large number of programs available in the Elliott library. These are not of course kept at Arups but if required could be obtained with a couple of days' delay. The great majority are of no interest to the structural engineer and few have been converted for use on a 4100 series machine (although this would not be an insuperable objection if the program were of real value). Those which might be of use to us and could be run unmodified on our own machine deal with such topics as traverse surveys, interpolation, the classical transportation problem, statistics and the analysis of experiments.

Elliott Program No. 33 cannot go unmentioned for this will translate integers into English, French, German, Italian or Swahili and into Roman figures. No one has yet modified this for our computer, perhaps because of the restrictions of the program, for numbers greater than 3999 cannot be converted into Roman figures while the largest number that can be translated into Swahili is 999,999.

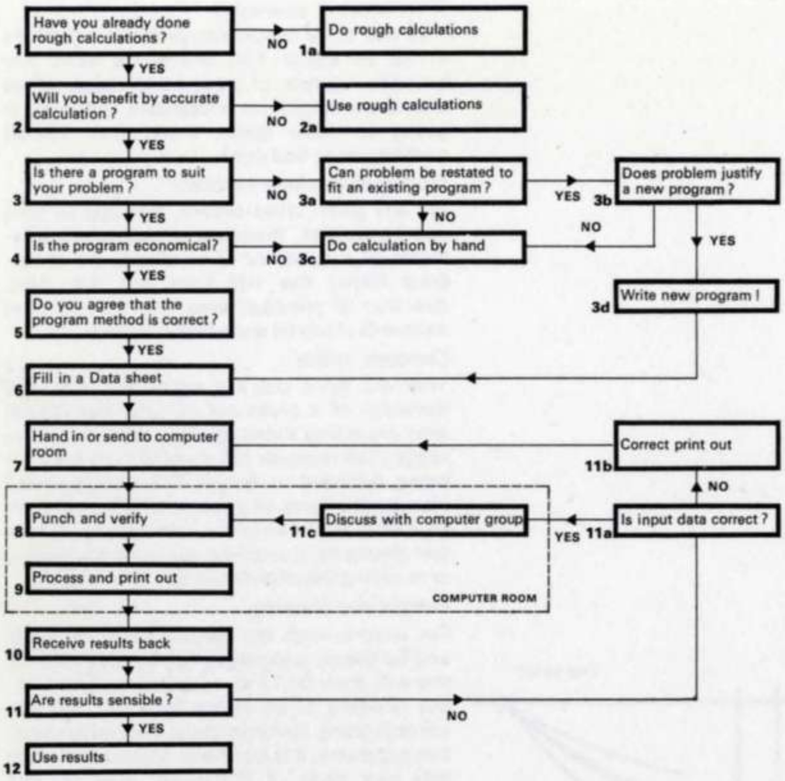


Fig. 1 Illustration of steps to be taken to use the computer

Using the machine

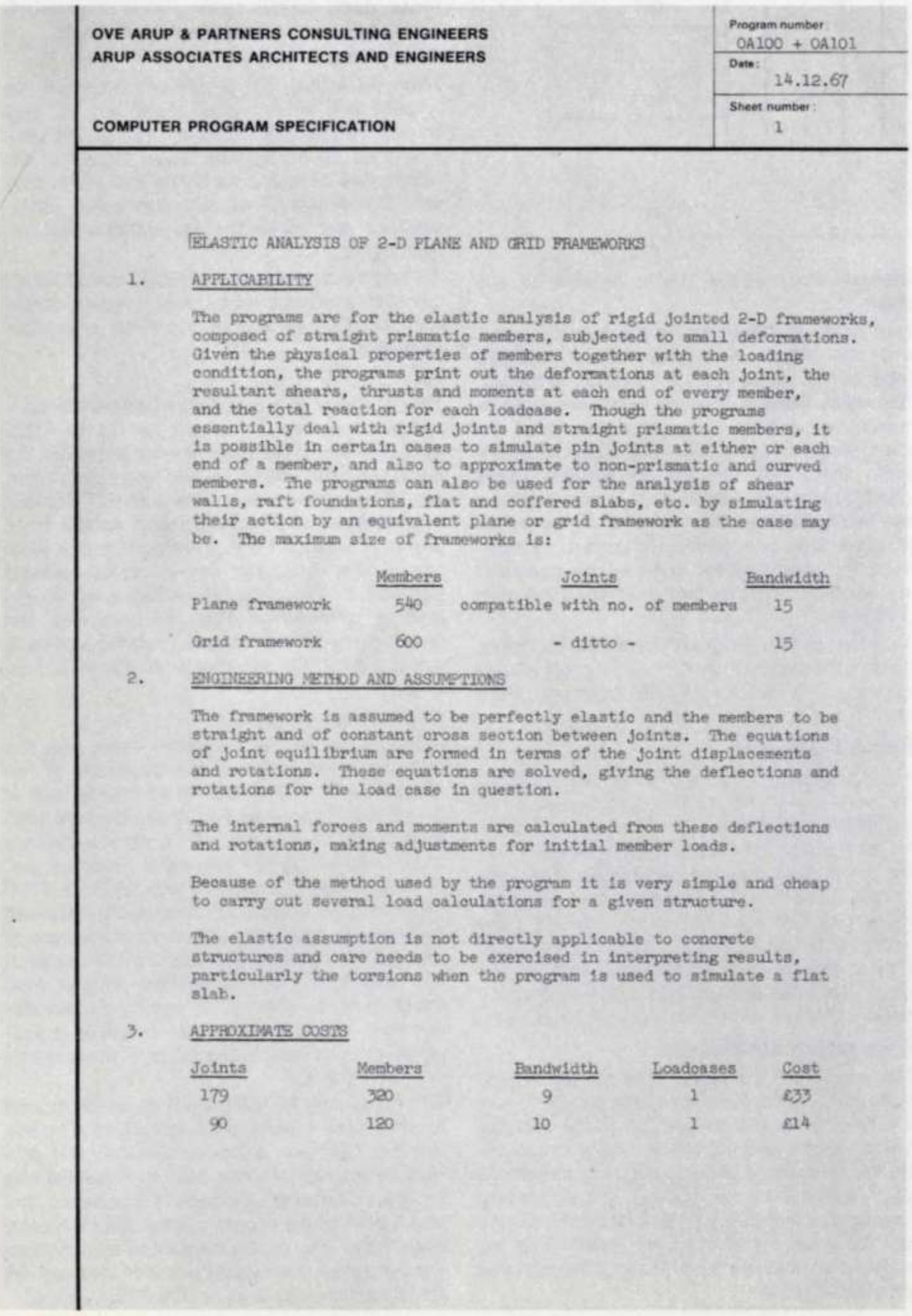
Charles Wymer

For our purposes the computer is usually just another way of doing sums. It is a central facility for doing sums in a way analogous to the Detailing Group. All we need to do is to describe the essential data of the problem (data sheet), indicate the way in which the problem is to be solved (the program to be used) and pass it on to the Computer Group.

The way to go about making use of the computer is indicated in diagram 1 and each of the steps is explained by the following notes. When you have once made use of the computer these steps will be self-evident and reference to the diagram or notes will be superfluous.

Machine time is quite expensive and its use is not justified where a simple calculation and approximate results will do. The diagram gives you a few questions to ask yourself, and a few people to consult before diving into the business of filling in data sheets.

If you have not yet made use of the computer it would be a valuable exercise to check each of your structural problems against the diagram until you find one that is suitable, and then use it. Suspend judgement of its value until you have made use of it for three or four problems.



OVE ARUP & PARTNERS CONSULTING ENGINEERS
ARUP ASSOCIATES ARCHITECTS AND ENGINEERS

COMPUTER PROGRAM SPECIFICATION

Program number: OA100 + OA101
Date: 14.12.67
Sheet number: 1

ELASTIC ANALYSIS OF 2-D PLANE AND GRID FRAMEWORKS

1. APPLICABILITY

The programs are for the elastic analysis of rigid jointed 2-D frameworks, composed of straight prismatic members, subjected to small deformations. Given the physical properties of members together with the loading condition, the programs print out the deformations at each joint, the resultant shears, thrusts and moments at each end of every member, and the total reaction for each loadcase. Though the programs essentially deal with rigid joints and straight prismatic members, it is possible in certain cases to simulate pin joints at either or each end of a member, and also to approximate to non-prismatic and curved members. The programs can also be used for the analysis of shear walls, raft foundations, flat and coffered slabs, etc. by simulating their action by an equivalent plane or grid framework as the case may be. The maximum size of frameworks is:

	Members	Joints	Bandwidth
Plane framework	540	compatible with no. of members	15
Grid framework	600	- ditto -	15

2. ENGINEERING METHOD AND ASSUMPTIONS

The framework is assumed to be perfectly elastic and the members to be straight and of constant cross section between joints. The equations of joint equilibrium are formed in terms of the joint displacements and rotations. These equations are solved, giving the deflections and rotations for the load case in question.

The internal forces and moments are calculated from these deflections and rotations, making adjustments for initial member loads.

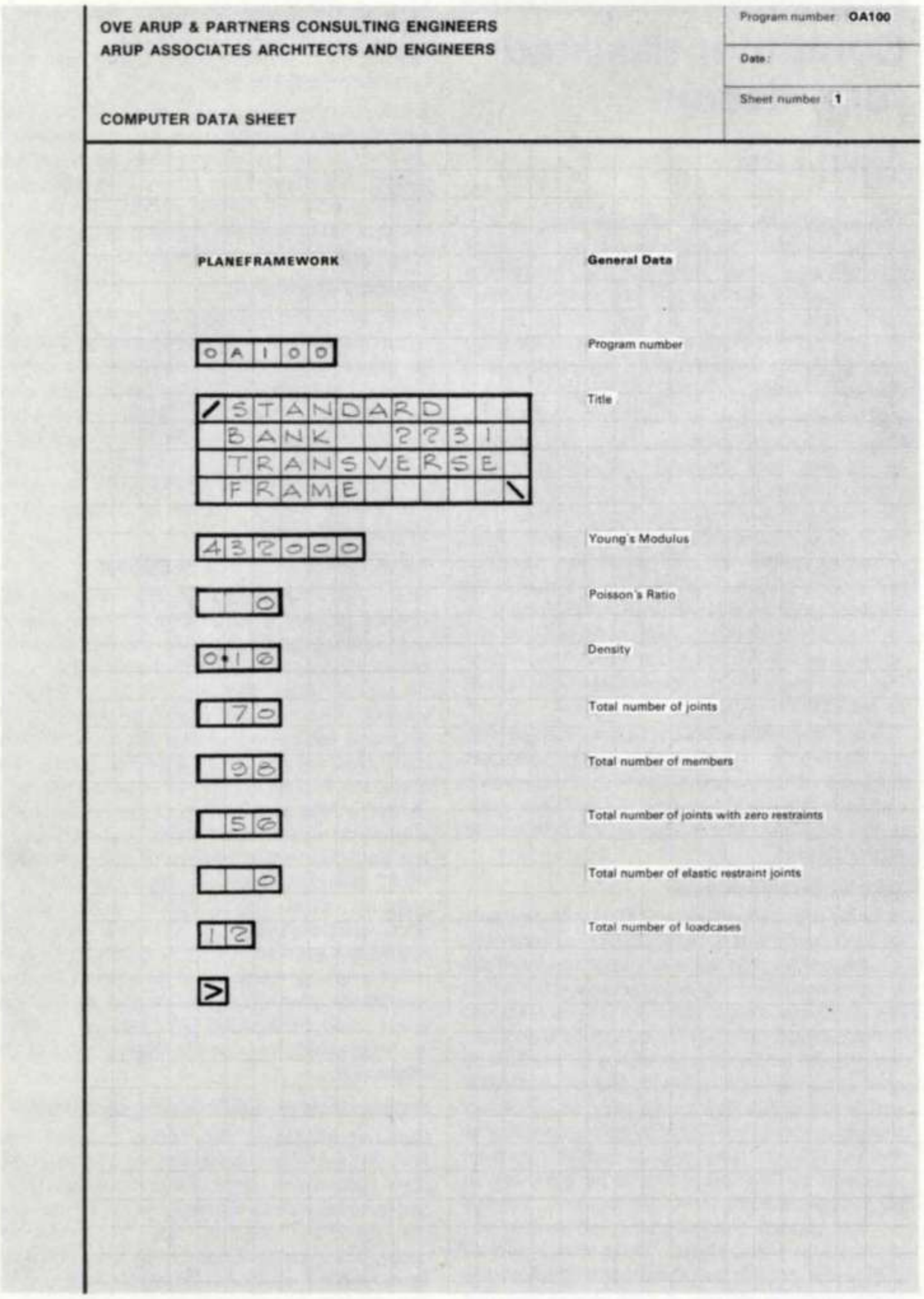
Because of the method used by the program it is very simple and cheap to carry out several load calculations for a given structure.

The elastic assumption is not directly applicable to concrete structures and care needs to be exercised in interpreting results, particularly the torsions when the program is used to simulate a flat slab.

3. APPROXIMATE COSTS

Joints	Members	Bandwidth	Loadcases	Cost
179	320	9	1	£33
90	120	10	1	£14

Fig. 2 Heading page of the program specification for the frameworks programs



OVE ARUP & PARTNERS CONSULTING ENGINEERS
ARUP ASSOCIATES ARCHITECTS AND ENGINEERS

COMPUTER DATA SHEET

Program number: OA100
Date:
Sheet number: 1

PLANEFRAMEWORK | **General Data**

OA100	Program number
/STANDARD BANK 2231 TRANSVERSE FRAME\	Title
432000	Young's Modulus
0	Poisson's Ratio
0·10	Density
70	Total number of joints
98	Total number of members
56	Total number of joints with zero restraints
0	Total number of elastic restraint joints
12	Total number of loadcases
>	

Questions

1.
Have you already done rough calculations? Before launching into machine calculation you should always have a rough idea of what the answer will be for two reasons—firstly to confirm that the proposed solution is valid and secondly to enable you to check the result.

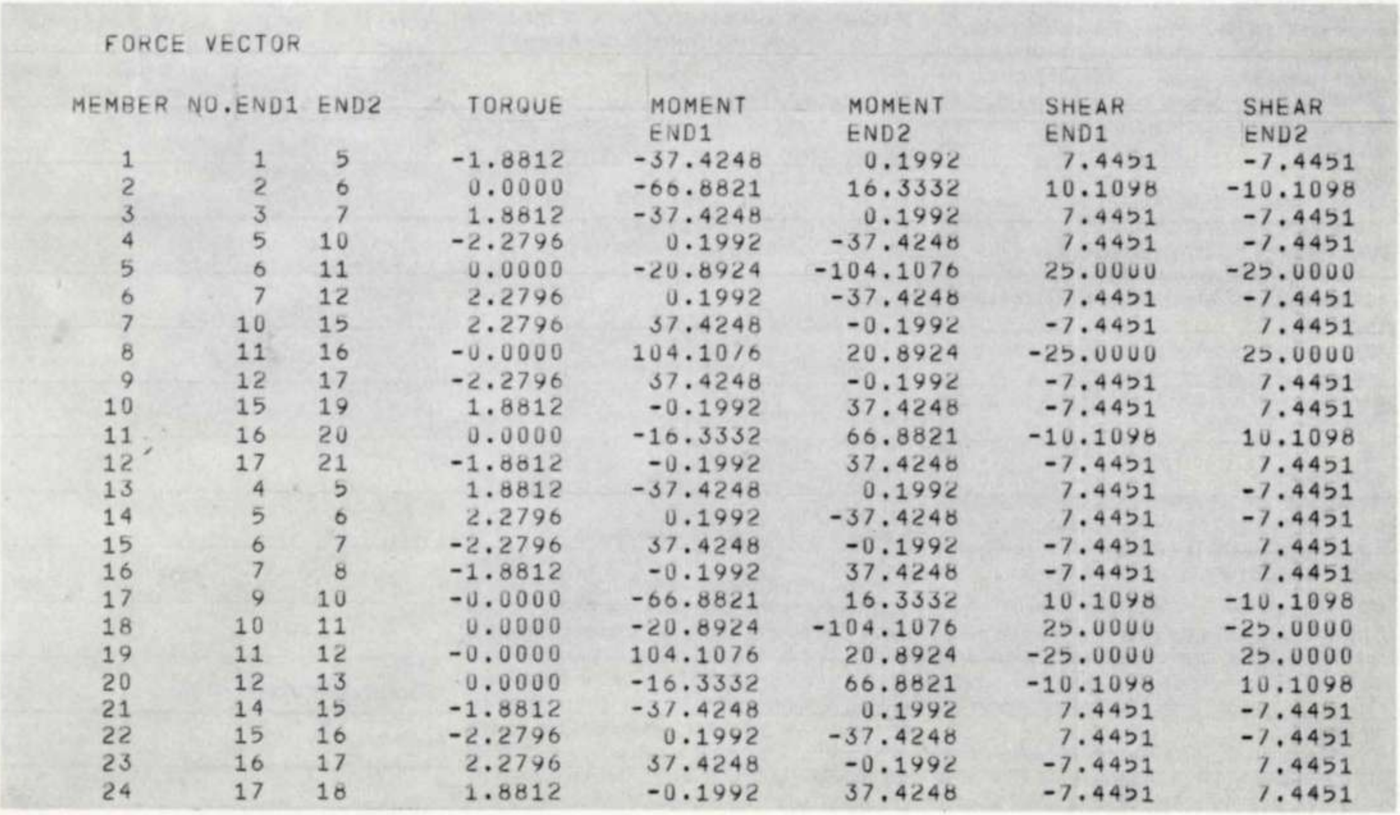
2.
Will you gain by accurate calculation? A little bit of simple arithmetic to give a rough answer to your problem is always quicker and cheaper than having the machine flogging away at £1 a minute. If your problem is not yet clearly defined, particularly in the early design stages, stick to rough calculations.

3.
Is there a program to suit your problem? Check through the book of program specifications kept by each floor secretary. The first page of the program notes defines the applicability of the program. (Fig. 2)

3A.
Can your problem be restated to fit a program? There may not be a program which self-evidently fits your problem. This may be because the program has a generalized title which sounds erudite, e.g. *'Structural analysis of 2-dimensional frameworks'*. This is, however, directly applicable to the common roof truss and especially useful if you want to take account of fixed joints or to find deflections. John Blanchard's article has some suggestions, or you can go and see someone in the Computer Group to help you to decide if a program exists to suit your problem.

Fig. 3 Heading data sheet for plane frame and grid frame programs

Fig. 4 Print out of results from plane frame work program



```
   FORCE VECTOR

MEMBER NO.END1 END2      TORQUE      MOMENT      MOMENT        SHEAR       SHEAR
                                     END1        END2          END1        END2
   1         1    5     -1.8812    -37.4248      0.1992       7.4451     -7.4451
   2         2    6      0.0000    -66.8821     16.3332      10.1098    -10.1098
   3         3    7      1.8812    -37.4248      0.1992       7.4451     -7.4451
   4         5   10     -2.2796      0.1992    -37.4248       7.4451     -7.4451
   5         6   11      0.0000    -20.8924   -104.1076      25.0000    -25.0000
   6         7   12      2.2796      0.1992    -37.4248       7.4451     -7.4451
   7        10   15      2.2796     37.4248     -0.1992      -7.4451      7.4451
   8        11   16     -0.0000    104.1076     20.8924     -25.0000     25.0000
   9        12   17     -2.2796     37.4248     -0.1992      -7.4451      7.4451
  10        15   19      1.8812     -0.1992     37.4248      -7.4451      7.4451
  11        16   20      0.0000    -16.3332     66.8821     -10.1098     10.1098
  12        17   21     -1.8812     -0.1992     37.4248      -7.4451      7.4451
  13         4    5      1.8812    -37.4248      0.1992       7.4451     -7.4451
  14         5    6      2.2796      0.1992    -37.4248       7.4451     -7.4451
  15         6    7     -2.2796     37.4248     -0.1992      -7.4451      7.4451
  16         7    8     -1.8812     -0.1992     37.4248      -7.4451      7.4451
  17         9   10     -0.0000    -66.8821     16.3332      10.1098    -10.1098
  18        10   11      0.0000    -20.8924   -104.1076      25.0000    -25.0000
  19        11   12     -0.0000    104.1076     20.8924     -25.0000     25.0000
  20        12   13      0.0000    -16.3332     66.8821     -10.1098     10.1098
  21        14   15     -1.8812    -37.4248      0.1992       7.4451     -7.4451
  22        15   16     -2.2796      0.1992    -37.4248       7.4451     -7.4451
  23        16   17      2.2796     37.4248     -0.1992      -7.4451      7.4451
  24        17   18      1.8812     -0.1992     37.4248      -7.4451      7.4451
```

3B.
Does the problem justify a new program? In certain cases where no program exists, the number of times the same problem has to be solved may justify a new program. Again the Computer Group should be consulted.

4.
Is the program economical? On the program notes you will find a summary of approximate costs with which you can decide if it is right to use the machine. You may wish to discuss this question with a group engineer.

5.
Do you agree that the program method is correct? As the engineer *you* decide if the assumptions and method used in the program are valid for your problem. You must under no circumstances take it for granted.

6.
Fill in the data sheet (Fig. 3). This is a straightforward operation and it is one that you do every week when you fill in a time sheet. Data sheets are kept by floor secretaries and in the Computer Room. Use a pencil so that you can erase and correct errors. Be careful to write clearly. The data will be read and put onto tape by someone who will not be able to recognize numbers by their context.

7, 8, 9.
Computer Room. These processes in the overall operation are the mystical bit. As far as we are concerned it could just as well be a group of people solely devoted to arithmetic. It is perhaps interesting but totally unnecessary to know how the arithmetic is done.

10.
Receive results back (Fig. 4). They will come on sheets of paper with holes down the sides about 24 hours after the data sheet has been sent in. The data that was fed into the machine is always printed out as a check and this is followed by the results. These could be incorrect.

11.
Are the results sensible? Check that the results are roughly what you expected. Check especially that equilibrium is satisfied. It is useful to express the results in some appropriate form, perhaps graphically, or as some other type of diagram that will quickly and obviously throw up any inconsistency. Never assume that results from the machine are correct.

11A.
Are the data correct? If you suspect an error in the results first look at the print-out of the data. Most errors occur here either because the data sheet was incorrectly or illegibly filled in (perhaps the units are incompatible) or less frequently because it was incorrectly punched onto tape.

11B.
Correct data on print-out. If you find an error in the data print-out, correct it and make sure that there are no others. Return this sheet to the Computer Room, so that the corrections can be made and a new set of results obtained.

11C.
Discuss errors. If there are no errors in the data but you are suspicious of the results take the print-out and discuss it with someone in the Computer Group.

12.
Use results.
When you are satisfied that the results are sensible then use them. Remember, however, that they are your results and your responsibility, not those of the Computer Group.

One last point

One of the reasons for getting the machine is that for many of our repetitive problems it is believed to be economical and it releases people to get on with the real problems of design.

If you have used programs unsuccessfully and are still drawing force diagrams for steel and trusses don't keep quiet about it, let us know.

Computer-assisted road design

Keith Law

The progressive stages of road location and design involve the engineer in calculations that are sometimes complex and sometimes just manually tedious but consistently time-consuming; this is the typical cue-line for the entry of the computer onto the scene, an entry that is being welcomed in the Roads and Bridges Division.

We are making use of the *Elliott 4100* Road Design Package which has been passed to us for testing and development under actual design conditions—a harder climate than it had previously been required to weather. The package is a suite of programs that attempts to automate some of the processes through which the engineer passes in the stages of road location and design but at the same time allows him to remain in full control of all the variable design factors, the programs providing him with quantified assessments of the design as it proceeds.

The comprehensive nature of the package may be judged from the Road Design Process diagram which seeks to show the relationship between the design stages and possible computer output at each stage to line printer and digital plotter.

Digital terrain model

In order that calculations of land requirements and earthworks may be made by the computer, it is necessary that the existing ground surface is defined sufficiently accurately in a form that the computer can assimilate. This is achieved comparatively quickly by inputting the existing ground levels of points on a defined 50 ft. grid network. Alternatively, the co-ordinates and levels of random points may be input and a separate program will interpolate levels onto the 50 ft. grid. The ground surface is then assumed by later programs to be a mosaic of 50 ft. square hyperbolic paraboloids, defined by their corners, thus allowing the level of any point to be interpolated. On a check test of 200 levels on one particular scheme involving quite irregular terrain, it was found that this method was sufficiently accurate for the purpose (the average error being of the order of 4 in. to 5 in.) and, what is more important, the errors cancel out.

Horizontal alignment

The horizontal alignment program accepts design input in the form of the co-ordinates of intersection points of straights together with the required radii and design speed for the linking curves that the computer will apply between these specified straights. From these data the program calculates the co-ordinates of tangent points (the points where curves meet straights) and of points at regular 50 ft. intervals along the centre line of the roads together with other relevant crossfall and direction data. In doing so, it will also calculate and apply the appropriate transition curves when required (transition curves are curves introduced between sharper circular arcs and straights in order to provide a gradual, rather than abrupt, change of curvature). Alternatively, it is possible to specify the lengths of transition curves, whereupon the program checks and displays the calculated length and warns if the specified length is insufficient. The program will not accept compound curves (i.e. curves compounded from a number of circular arcs of the same hand with linking transition curves) nor will it continue to run should adjacent curves overlap on the joining straights. As it is our policy to aim for free flowing, continuously curved design with the absolute mimimum use of straights, it will be understood that these two disadvantages are quite serious and it is hoped that present development work on the program will eliminate them in the future.

Separate programs are used for the output of information which may be either in tabular form on the line printer, or graphical on the digital plotter as a plan of the centre line, or in conjuction with the terrain model, as a longitudinal section of the existing ground along the proposed road centre line.

Vertical alignment

From an input defining the principal constituent gradients of the vertical alignment the computer calculates the radii for the vertical curves (checking as it does so that adequate visibility is provided over 'humps') and works out the channel levels at 50 ft. intervals along the road. These levels are stored on magnetic tape and may be output in a tabular form on the line printer or be plotted as a longitudinal section of the road.

Calculation of cross sections

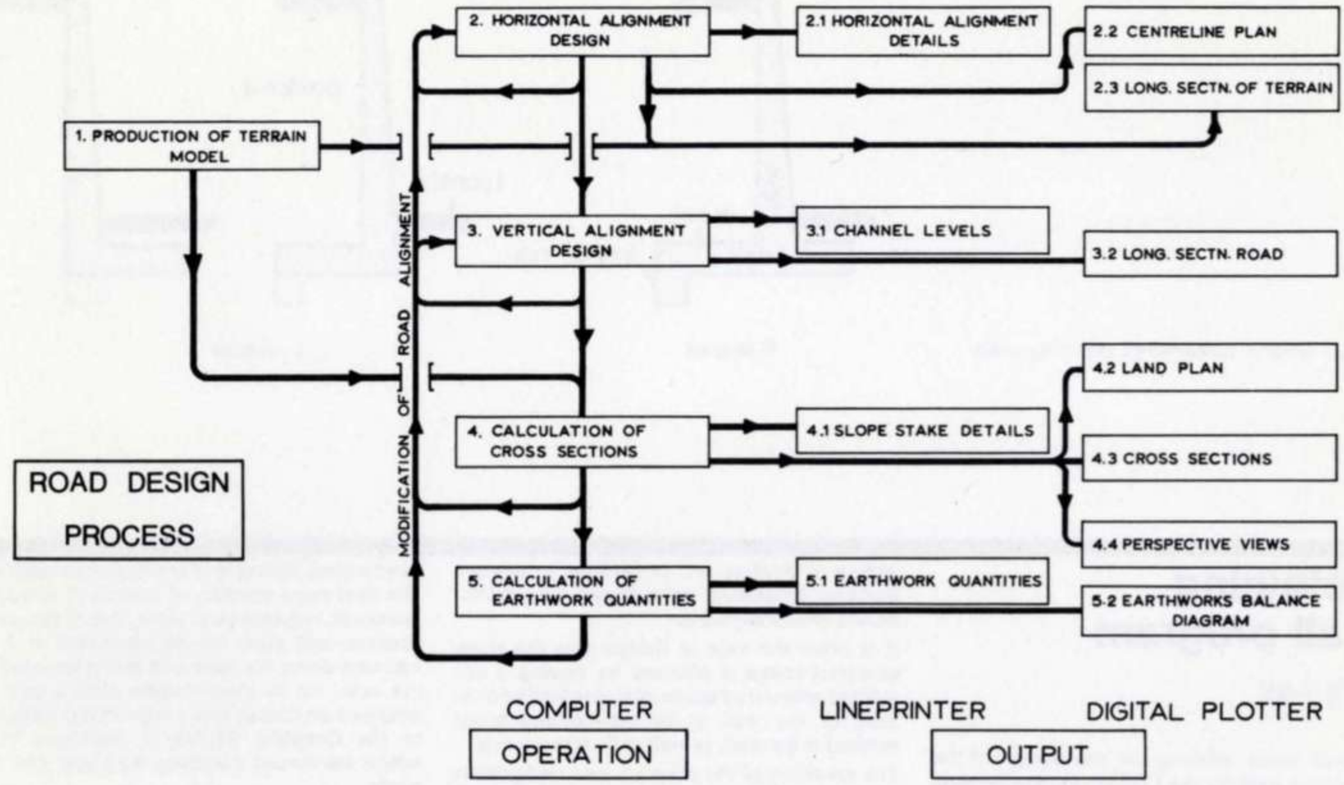
Having thus far decided on the horizontal and vertical alignment to be used, it is now necessary to design and specify to the computer the dimensions and slopes of the constituent parts of typical cross sections of the road (e.g. widths of verges, carriageways, slope of cutting face, etc.) and to define the lengths of the road over which such sections apply. The computer is then programmed to accept such 'patterns' and apply them to the chosen road alignment and terrain model, thus calculating the particular cross section of the road every 50 ft. along its length, again to be stored on magnetic tape. From tape it is possible to either tabulate the offsets from the road centre line to the extremities of the cutting/embankment slope, or carry out a number of plotting operations graphically, as shown on the diagram, such as drawing perspectives to assist in design decisions on the appearance of the alignment.

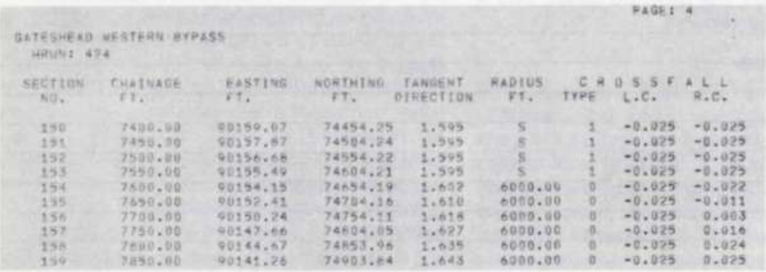
Calculation of earthworks quantities

Previous stages of the design process have fully defined the existing terrain and proposed road surfaces in three dimensions and cross sections have been calculated. No further data are therefore required for all earthworks quantities to be calculated onto magnetic tape from where they may be fully tabulated on the line printer (as shown in the example) or plotted as an earthworks balance diagram. There is no provision as yet in the programs for defining a secondary surface (such as rock) at a lower level, or for the calculation of such separate quantities. It is hoped that this may be one of our future developments.

It will be appreciated from this brief outline that the computer can be and will become increasingly useful and powerful in its partnership with the road engineer. It allows him the opportunity and time to make detailed comparisons of costs and appearance between a proper number of route alternatives, where previously he was probably restricted to one or two. By removing the time-consuming drudgery from highway design, the computer is enabling the engineer to exercise his proper professional duties more fully than ever before.

Right:
Gateshead Western by-pass
Road design process diagram
and below it
typical sections of output



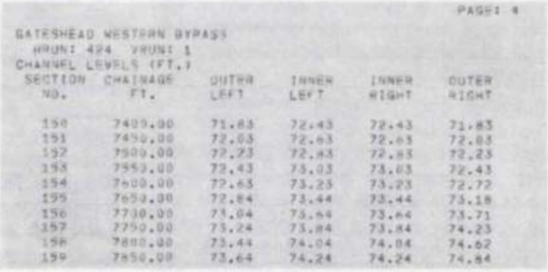


```
                                                                    PAGE: 4
GATESHEAD WESTERN BYPASS
  HRUN: 424

 SECTION   CHAINAGE     EASTING    NORTHING  TANGENT     RADIUS    C R O S S F A L L
   NO.       FT.         FT.          FT.    DIRECTION    FT.    TYPE  L.C.     R.C.

   150      7400.00    90159.07    74454.25   1.595       S         1   -0.025  -0.025
   151      7450.00    90157.87    74504.24   1.595       S         1   -0.025  -0.025
   152      7500.00    90156.68    74554.22   1.595       S         1   -0.025  -0.025
   153      7550.00    90155.49    74604.21   1.595       S         1   -0.025  -0.025
   154      7600.00    90154.15    74654.19   1.602    6000.00      0   -0.025  -0.022
   155      7650.00    90152.41    74704.16   1.610    6000.00      0   -0.025  -0.011
   156      7700.00    90150.24    74754.11   1.618    6000.00      0   -0.025   0.003
   157      7750.00    90147.66    74804.05   1.627    6000.00      0   -0.025   0.016
   158      7800.00    90144.67    74853.96   1.635    6000.00      0   -0.025   0.024
   159      7850.00    90141.26    74903.84   1.643    6000.00      0   -0.025   0.025
```

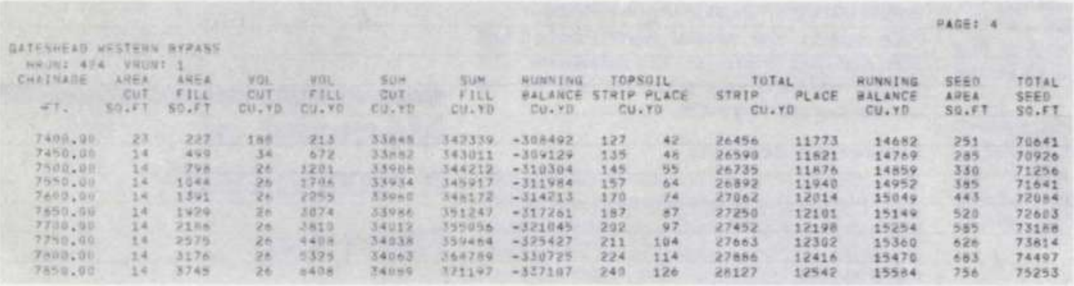
2. 1 Horizontal alignment details



```
                                                          PAGE: 4
GATESHEAD WESTERN BYPASS
  HRUN: 424  VRUN: 1
CHANNEL LEVELS (FT.)
 SECTION  CHAINAGE    OUTER     INNER     INNER     OUTER
   NO.       FT.      LEFT      LEFT      RIGHT     RIGHT

   150     7400.00    71.83     72.43     72.43     71.83
   151     7450.00    72.03     72.63     72.63     72.03
   152     7500.00    72.23     72.83     72.83     72.23
   153     7550.00    72.43     73.03     73.03     72.43
   154     7600.00    72.63     73.23     73.23     72.72
   155     7650.00    72.84     73.44     73.44     73.18
   156     7700.00    73.04     73.64     73.64     73.71
   157     7750.00    73.24     73.84     73.84     74.23
   158     7800.00    73.44     74.04     74.04     74.62
   159     7850.00    73.64     74.24     74.24     74.84
```

3. 1 Channel levels

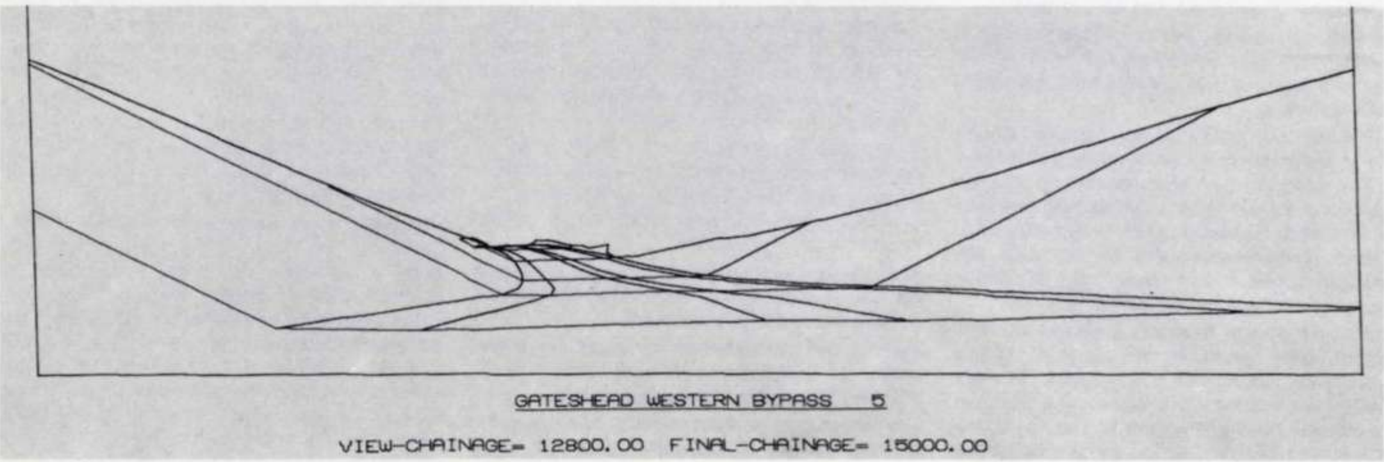


```
                                                                                                      PAGE: 4
GATESHEAD WESTERN BYPASS
  HRUN: 424  VRUN: 1
 CHAINAGE   AREA   AREA    VOL     VOL     SUM      SUM    RUNNING   TOPSOIL          TOTAL        RUNNING   SEED   TOTAL
             CUT   FILL    CUT     FILL    CUT      FILL   BALANCE STRIP PLACE   STRIP     PLACE  BALANCE   AREA   SEED
   FT.     SQ.FT  SQ.FT   CU.YD   CU.YD   CU.YD    CU.YD    CU.YD    CU.YD          CU.YD          CU.YD    SQ.FT  SQ.FT

  7400.00    23    227     188     213    33848   342339  -308492   127    42    26456     11773    14682    251   70641
  7450.00    14    499      34     672    33882   343011  -309129   135    48    26590     11821    14769    285   70926
  7500.00    14    798      26    1201    33908   344212  -310304   145    55    26735     11876    14859    330   71256
  7550.00    14   1044      26    1706    33934   345917  -311984   157    64    26892     11940    14952    385   71641
  7600.00    14   1391      26    2255    33960   348172  -314213   170    74    27062     12014    15049    443   72084
  7650.00    14   1929      26    3074    33986   351247  -317261   187    87    27250     12101    15149    520   72603
  7700.00    14   2186      26    3810    34012   355056  -321045   202    97    27452     12198    15254    585   73188
  7750.00    14   2575      26    4408    34038   359464  -325427   211   104    27663     12302    15360    626   73814
  7800.00    14   3176      26    5325    34063   364789  -330725   224   114    27886     12416    15470    683   74497
  7850.00    14   3745      26    6408    34089   371197  -337107   240   126    28127     12542    15584    756   75253
```

5. 1 Earthwork quantities

TYPICAL
SECTIONS
OF
OUTPUT



4. 4 Perspective view

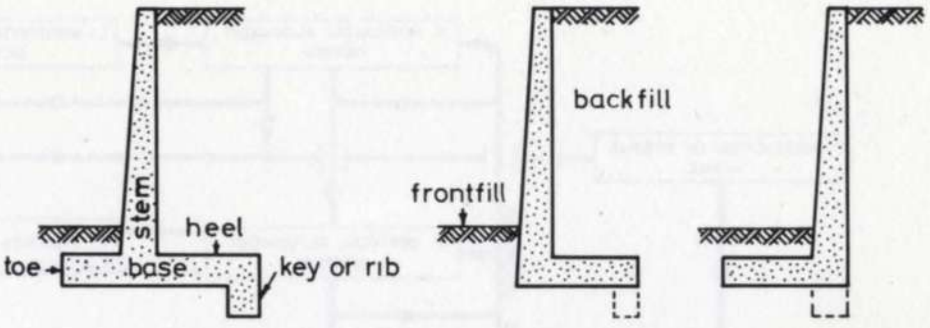


Fig. 1 Simple cantilevered retaining walls

T-shaped

L-shaped

Retaining wall program

Bill Hill

It was while working on the design of the retaining walls for the 1,000 ft. long underpass in Watford that I first became involved in the development of a computer program which will design or analyse simple reinforced concrete cantilevered retaining walls such as shown in Fig. 1.

Due to the variation in height (between 0 and 20 ft.) of the underpass retaining walls, it was necessary to design a number of different wall sections. Anyone who has designed a retaining wall will realise that to design one section involves at least two trials to arrive at the optimum base length with the heel and toe proportioned such that the stability and bearing pressure requirements are satisifed.

We had reached the stage where we had plodded through 70 or more pages of preliminary calculations using anticipated soil properties for the stability calculations. The need for some calculating machine at this stage was apparent but not foremost in my mind.

Shortly after completing these calculations site investigation and subsequent laboratory tests were carried out on the soil in the vicinity of the underpass. It then became apparent that the calculations would have to be repeated, unfortunately by me.

Dawning light

The prospect of this sent me scurrying down to the basement of No. 8 to see if we had got a program available for such repetitive calculations or if we could possibly write one.

As it happened although we did not have such a program I was informed that Maunsell & Partners had and that it might be possible for us to borrow it.

It was not long before we had acquired a copy of the Maunsell program and duly scrutinised it. The program had been written to choose the best dimensions for a simple non-tapering cantilevered retaining wall complying with stability requirements and to calculate the bending moments and shear forces at various positions on the stem and base of the wall.

Unfortunately we found that certain modifications were necessary to the logic of the stability calculations in the program. In view of this we agreed with Maunsells that we should rewrite the program and at the same time extend it to include the calculation of the required steel areas corresponding to the calculated bending moments, the possibility of tapered stems, the option of being able to design or analyse and to be able to impose practical limitations on wall and base thicknesses and dimensions.

It is often the case in design that the cross sectional shape is affected by having a restricted amount of space available behind or in front of the wall or by the constructional method to be used, as well as by appearance.

The rewriting of the program was undertaken by David Taffs who in spite of my help managed to break the program down into a series of logical procedures.

General program

During this time we had meetings with Maunsell, Sir Alexander Gibbs and eventually Freeman Fox & Partners, to discuss the general requirements of a retaining wall design program. These meetings resulted in an agreed format for a general program which would cater for all types of cantilevered retaining walls (without counterforts) including walls forming part of buildings and bridge abutments.

It was decided that our revised form of the Maunsell program, when completed, should serve as a basis for the final program, which would be written by us in the near future.

Two months later, several months since I had first scurried down to the basement, the revised program was ready for use for the underpass retaining walls.

The revised program

The alternative of being able to use the program for design or analysis proved very useful. We first designed several wall sections varying in height from about 3 ft. to 20 ft. From these results, bearing in mind possible positions of construction and expansion joints it was possible to decide on a reasonable number of changes in section required along the underpass.

Suitable sections, perhaps being modified versions of the above designs, were then chosen for analysis. The analysis checked bearing pressures, factors of safety and determined the reinforcement required.

It was also possible to see the effect of using different soil properties on the wall section, as even with the familiar cry of 3 tons/sq. ft., O = 30°, from the third floor no. 8, these properties are not absolute.

The program was found to take approximately 30 seconds to design or analyse one wall section and output all the results.

Results are conveniently arranged on three sheets of output for each wall section such that they can be cut to A4 size and filed.

The first page of output consists of a comprehensive list of all the input data for easy reference. The second page consists of all wall, base and key dimensions, bearing pressures and factors of safety against the occurrence of overturning, sliding and shallow shear failures. The final page consists of a table of bending moments, required steel areas, actual concrete stresses, and shear forces calculated at 2 ft. intervals down the stem and along the base of the wall. Up to three carbon copies can be obtained on output and a copy could be given to the Detailing Section if necessary from which reinforced concrete drawings can be made.

At one stage early in the development of the retaining wall program, the possibility of taking the program to its logical conclusion to produce a bar bending schedule and estimate of material quantities was discussed. This would have involved the standardisation of a system or alternative systems of detailing for all possible wall cross-sections. Even so, there would be exceptional cases to consider. In the interests of everyone concerned, it was thought better to have a program which could bark in the near future rather than be continually developing one which would eventually be able to sing.

Memoirs of a troglodyte?

David Taffs

'How would you like to spend some time in the basement?' the man said. There were the undertones of a threat. For 'spend some' insert 'do', I thought. It was November 1966 and the fact that I had worked for three different groups in a space of two years with Arups took on a new significance. For which of my many sins was I now to answer? Perhaps one of my frameworks had at last manifested itself as a mechanism, or a yield line not possessing my keen sense of direction had taken a wrong turning.

My fears were temporarily allayed when I heard that the object of the move was to produce a library of computer programs for structural element design. The idea of working continuously for more than one year with the computer appealed to me as much as it would to most engineers. Nevertheless, two factors eventually tipped the scales in favour of the move. One was David Lowes' prediction of a hinge forming in the G.P.O. Tower, the other was the potential gratuities from directing the partners in and out of No. 8's car park.

I began my indoctrination in December 1966 with the intention of producing a bevy of programs by the following summer. Much of my initial time was spent familiarising myself with the system, studying the computer manuals and assimilating the finer points of *Algol* programming. During this introductory period I was developing two preliminary programs, one on column loading and the other on column design. These were not intended for general use but for adaptation and incorporation into a larger and more general program. Some 12 weeks had passed before I put aside these projects and it was during this time that I began to appreciate why there were so few working programs on the market, despite the many teams of people that were employed in the computer industry developing structural engineering programs.

Following the initial skirmishes with the computer I settled down to the task of producing programs for use within the firm as aids to structural design. It was my aim to produce programs that were not hampered by those annoying restrictions which are so often encountered by the designer when he attempts to use a computer. His structure seldom seems to fit the program and it is usually the former that ends up being adapted in order to get an approximate solution.

All the programs I am working on are being written with the intention that they will be used at two stages. The first stage is when the designer is in the process of sizing the structural members. Not all members are determined by architectural or similar factors but limitations to possible sizes are known and when using the programs these limitations are specified, whereupon the computer determines a size of member within the specified range. Slight variations in size between members can then be rectified by the designers. The second stage in the use of the program comes when the final design is required. At this point all member sizes should be known so an analysis of them is required in order to arrive at steel details.

Element design

The possible topics within structural element design that could be developed were innumerable but a column base seemed a logical place to start and, in addition, Colin Davies had produced an experimental program that designed and detailed for a concentric load. I spent approximately three weeks investigating the techniques of column base design and evolving the relevant equations having accepted that the premise of a perfectly elastic soil was sufficiently accurate, another three weeks accounted for flow charting. Debugging the program, i.e. finding and correcting the mistakes in the syntax, logic and mathematics and preparing extensions and draft code of practice requirements have so far taken seven weeks.

As the result of a request by Bill Hill we rewrote and extended a retaining wall program as explained in his article. The preparation of equations involved two weeks, writing the programs took one week while debugging has taken six. Extensions to this program to include a wider range of wall shapes and loadings have taken one week to prepare, one week to write and four weeks debugging to date.

A column and beam analysis program was an extension to the base design and involved similar equations. One week was spent writing it followed by four weeks of debugging. Development of each program proceeded concurrently and so the times quoted are an estimate of the total spent on each project. These times may appear surprisingly long but the difficulty of programming lies not in the complexity of the mathematical equations employed but in their organisation and evolving the structure of the decisions that follow a set of equation results.

A design program contains far more decisions than one that caters for analysis only. This is because the program must contain the logic that a designer would apply when altering the shape or size of a member. Only after giving the subject a lot of thought can one begin to appreciate how involved this becomes. A program that provides a solution for one particular case is very different to a general one that caters for many different cases.

A common difficulty encountered in the general program occurs when values become equal to zero. Many equations then become unnecessary but the machine, being ignorant of this, continues to plough through them, often ending in a number that is too large for the machine to hold or else is extremely small and, although virtually equal to zero, fails to satisfy a conditional statement that compares the value with zero.

Future aims

The programs I have mentioned may be extended and refined almost indefinitely but a compromise between cost of development and potential saving in labour has to be reached. A major problem that must soon be solved is that of feeding an entire structure into the machine where it can be stored as a complete record. Revisions to the record could be made at any times convenient to the designer. When requested, quantities of materials could be generated from stored information or analysis programs run by the computer automatically extracting relevant data.

My own ambitions include finding a computer that will not corrupt my programs, mispunch my tapes, improvise on my outputs, misread my inputs, read the data tape three times before accepting it, give up in the middle of a program, tear paper tape in the readers, lose its place on the magnetic tapes, forget it has magnetic tapes, have ineffective control buttons, have drum plotters that crumple the paper while slewing or pens that snag the paper while drawing...

There is an adage that comes to mind. Isn't it something about a workman and his tools?

The future of our computer

Poul Beckmann

Having exhaustively covered programs under development and even one or two which are still only intended, John Blanchard has not left me much to say about the future.

Predictions

I'm however pretty sure that we will still have people with more degrees than sense using the machine where some thought and half a page of A4 would have been much better.

We will also (pace Srinivasan!) get complaints that the machine will not accommodate bigger structures.

Furthermore as faith in the oracle increases in inverse proportion to knowledge of how structures work there will be an increasing amount of analyses carried out based on academically correct but factually irrelevant assumptions, leading to bigger sections and heavier reinforcement than is really necessary.

And even where the machine may be used with discretion the mere mass of information which the machine prints out will make it difficult to see the wood for the trees.

One way and another, unless by incessant instruction, by all the means at our disposal, we ensure that the machine is looked upon as just another calculating device, we shall find ourselves designing structures which are more expensive for our clients as well as for ourselves. The consequences of such a trend need little elaboration, bankruptcy is within sight.

The remedies

If, on the other hand, we could take advantage of the computer's capacity to perform repeating routine operations swiftly and reliably we could give our engineers far more time to think about what the problems confronting them really are.

Likewise, if in the cases when extensive computer analysis is necessary it could be arranged that a potted version of the output were produced, the engineer would get a much quicker appreciation of what goes on. This would be even more so if the plotter were made to draw bending moment diagrams, etc.

One of the first tasks for our programmers will therefore be to modify the existing frameworks programs so as to make the machine combine the various load cases and eventually to enable it to pick out the worst combinations.

This last operation may well be very difficult as one of the shortcomings of computers is their inability to make an intelligent choice. They have to be given rules by which to choose. It may be very difficult to frame a complete set of rules so one will occasionally question the wisdom of the machine's choice. That means that the individual load case results must be examined and they should therefore be available from a cold store in the form of magnetic tape or paper tape.

Curved members have up to now been dealt with as series of short straight ones. This becomes impractical when they form part of a grillage so the grid program needs to have a package inserted to enable it to deal with curved members directly and hence allow the Roads and Bridges Division to go round the bend.

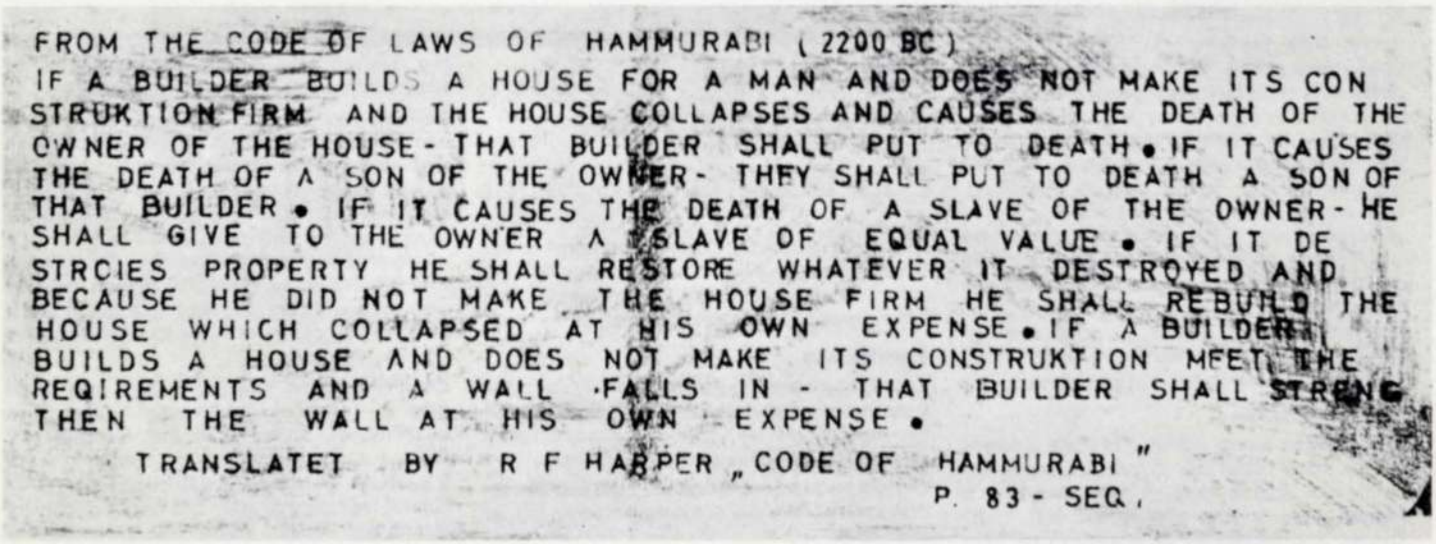
Whilst on this subject, cellular box structures require further development of the finite element techniques to facilitate their analysis.

Curves, whilst pleasing to the eye, can be very difficult to set out, especially for overhead roads where it is not practical to adjust the formwork on site. The experience from Sydney Opera House shows that it is feasible to give 90% of such setting-out information in the form of computer output, and this may well be more common in the future.

A field in which we have so far made little use of the machine is the preparation of design tables and graphs and standard solutions to standard problems to be used in preliminary design.

It has been argued and still is that our structures are not on a regular grid, etc. etc. It usually turns out that they are not all that far from something that can be simulated by a regular structure for a large part. What then is the point of each engineer working out his coffered slab afresh during the early design stage, when he could interpolate between two standard solutions from 'the album' and get his quantities near enough in next to no time? The more so as he could then look at the real question: 'Should it be a coffered slab?'

All in all, it therefore seems that the greatest reward we can get from using the machine in the future is relief from the tyranny of figures. This will, however, only come about by concentrating our resources on the surprisingly difficult task of programming the machine to do elementary work, and to give us the answers in a simple and comprehensible form.



FROM THE CODE OF LAWS OF HAMMURABI (2200 BC)

IF A BUILDER BUILDS A HOUSE FOR A MAN AND DOES NOT MAKE ITS CONSTRUKTION FIRM AND THE HOUSE COLLAPSES AND CAUSES THE DEATH OF THE OWNER OF THE HOUSE - THAT BUILDER SHALL PUT TO DEATH • IF IT CAUSES THE DEATH OF A SON OF THE OWNER - THEY SHALL PUT TO DEATH A SON OF THAT BUILDER • IF IT CAUSES THE DEATH OF A SLAVE OF THE OWNER - HE SHALL GIVE TO THE OWNER A SLAVE OF EQUAL VALUE • IF IT DESTRCIES PROPERTY HE SHALL RESTORE WHATEVER IT DESTROYED AND BECAUSE HE DID NOT MAKE THE HOUSE FIRM HE SHALL REBUILD THE HOUSE WHICH COLLAPSED AT HIS OWN EXPENSE • IF A BUILDER BUILDS A HOUSE AND DOES NOT MAKE ITS CONSTRUKTION MEET THE REQIREMENTS AND A WALL FALLS IN - THAT BUILDER SHALL STRENGTHEN THE WALL AT HIS OWN EXPENSE •

TRANSLATET BY R F HARPER „CODE OF HAMMURABI"
P. 83 - SEQ.