THE TRIDIMENSIONAL THERMOELASTIC COMPUTER CODE "TITUS"

P. LAUNAY, G. CHARPENET, C. C. VOUILLON,

Compagnie Industrielle de Travaux, Paris, France

ABSTRACT

The TITUS system is a program package permitting processing on a computer of a wide range of problems in the fields of theory of elasticity, strength of materials, thermic and thermoelasticity. This memorandum gives the basic options chosen, wether on the level of theory adopted, processing methods, programming or functional organization on computers Stress is placed on the ease of using the system, obtained by the great flexibility of inputs and outputs, and organized so that TITUS constitutes a tool with many possibilities in so far as both computations and dialogue with the user are concerned. Examples of structures designed at CITRA with the TITUS system are given as examples.

1. CONCEPT OF A STRUCTURAL DESIGN SYSTEM

The use of the finite element method started in our company in 1964, to study the structural behaviour of prestressed concrete pressure vessels. It resulted in a first axi-symmetric thermoelastic computer code which has been described in other papers [1] - [4].

A next step was to develop a program intended to solve most of the problems existing in designing unusually shaped plates as found in interchange highways structures. This program was called PLAQUE.

The increase in the number of program users resulted in the spontaneous appearance of new requirements. These requirements may be listed on three levels :

- On the design facility level : 3-dimensional structures, shells, combined plate beam structures
- On the level of program facility use : structures that can be designed on a computer become very complex. While this enables the users to grasp the schematization of their structures in all its details, this also results quickly in a large volume of data to be processed, both at input (introduction of geometric coordinates of a space meshing for instance) and at output (the results of an average design quickly take on vast proportions and their interpretation has a tendency to put the user off). Large volume always involves a risk of error. At this level, therefore, data checking procedures, automatic meshing, results interpretation procedure and finally, graphic outputs must be developed.

- On the level of program concept : both users and analysts have felt the necessity of unifying all the design options and of organizing and merging the processing procedures so that all their problems are treated by a single convenient tool.

In addition, the way of looking at the finite element method has become transformed In the beginning, this method was looked on as a way of cutting up a structure into separate pieces, this assuming that all parts would be identical in nature. To-day, a structure is considered rather as the result of assembly of pieces, these pieces being, a priori, of any nature. This new simple way of reasoning permits the design of combined structures consisting of components as different as beams associated with solids and/or with shells. With the former concept, the beams had to be meshed, which obviously reduced considerably the value of the method.

At the same time, in connection with these requirements, data processing resources have also evolved :

- in hardware (faster central memory, high-capacity memory with direct addressing,appearance of first digital plotters)
- in software (multiprogramming, dynamic memory management . . .)
- in programming technique (modular programming).

The synthesis of all these elements (user's requirements, evolution of concepts and data processing techniques) has gradually arisen the idea of a coherent organized and functional assembly : the TITUS system.

In this connection, two remarks should be made : firstly, the system prepared by CITRA has been influenced by the environment of designers; it is, first of all, a tool which must be versatil and practical to use by project engineers. This constant guide line has led to the development in priority of : input facility (free-format data, automatic plotting of introduced meshing, error detection, automatic generation of meshing. . . .)

: processing facility (control language, communicability among several computers)

: output facility (result storage, combination of loading cases, plotting of strained structure, internal efforts variation curves, principal stresses, equal values).

Secondly, while with a monolithic program, it is necessary to terminate the program completely in order to be able to use it, in a modular system like TITUS, once the general framework is determined, the service possibilities of such computing option are independent of the progress state of the other programs of the system. Furthermore, the number of modules is not limited and may be extend as new requirements arise.

Presented in this way, TITUS has four groups of functions (see fig. 1) :

- a general processing procedure managing all the operations
- specific design programs adapted to each type of problem encountered
- UPSTREAM programs permitting these problems to be defined

- DOWNSTREAM programs (post processing of results), helping the user to interpret and process the data obtained previously.

Acknowledgement must be made here to E D F (Electricité de France) and C E A (Commissariat à l'Energie Atomique):both organizations brought financial and technical support to our Company to build up one master piece of the TITUS system namely the tridimensionnal chapter. This work was performed in connection with a broader program of research and development on prestressed concrete reactor technology carried out with S E C N (Société d'Etudes des Caissons Nucléaires).

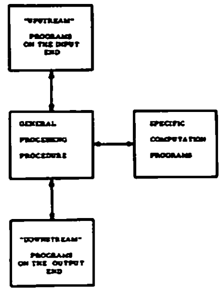


FIG 1. GENERAL FUNCTIONS OF THE "TITUS" SYSTEM

2. GENERAL PRESENTATION OF THE TITUS SYSTEM

The previous chapter has defined the different functions performed by the TITUS system. Let us examine them in greater details and note how they permit to satisfy all levels of requirements of the engineer.

2. 1 - PROCEDURES ON THE INPUT END ("UPSTREAM" PROGRAMS)

The engineer wishing to submit a study to the computer must first :

- schematize his problem
- communicate the corresponding data to the computer

The schematization is necessary in order to place the problem within the framework allowed by the program and to be able to introduce the data under a form that is standardized to a certain extent, i. e. comprehensible for the program. In fact, this schematization appears under several aspects :

- geometric schematization, schematization of physical characteristics
- schematization of bonds (supports . . .)
- schematization of loads.

Once this schematization is made, the communication of the corresponding data to the computer is done by means of "upstream" procedures. They define the problems to be processed, check partially and store data in the machine. They, therefore, play a "conversational" part versus the user. It is essential that they facilitate to the maximum data preparation, which may be a considerable work (3-dimensional structures for instance). Two procedures exist at present : use of printed forms and free-format clear language.

2. 1. 1 - Standard printed forms

These forms can satisfy all foreseen possibilities. In fact, they constitute practically a stored data file image. They are common to all computing options (only a few of them differ by their headings, in order to facilitate their use).

While they are relatively practical for small problems, these models become hard to use for complicated problems. Nevertheless, they have the advantage of forming a simple data input basis, permitting to benefit from all computing facilities.

2. 1. 2. - Data input language

The use of standard printed forms involves the constraint of centring, exacting for the user and often being a source of errors, In addition, it requires data to be introduced under a coded form, often hard to assimilate. Lastly they are poorly adapted to the choice possibilities of different procedures. This choice is necessary within a general system, inevitably involving excessive model proliferation. An uncoded data input language consisting of phrases and numbers satisfies much better the requirements of a system such as TITUS. Its easy syntax and its vocabulary very close to the current jargon of the profession, make it very easy to assimilate by the engineer. Its obvious signification facilitates re-reading and checking, and the absence of centring constraints eliminates most punching errors. In addition the description of the structure to be designed follows closely the step of the engineer when he sets the problem to himself. Lastly, the choice of desired procedures is done naturally by the use of suitable words and phrases without necessitating any other aid than a sufficiently clear user's manual.

DEFINITION	- This control card specifies that all the following ones (until a RETOUR card is read) are input data cards.
DUPONT/PLAQUE SUR POTEAUX/N° 1243	- Name/Title/References
OPTION PLAQUE	- The calculation will be completed later using option number 2 (plates)
RECTANGLES	
X = 2. Y = 1.5 I = J = 5	
CARACTERISTIQUES PHYSIQUES	- Physical charactéristics (thickness, density, modulus of elasticiy Poisson's ratio)
H = 0.25 RO = 2.5 E = 1 500 000 NU = 0.15	
LIAISONS	- Bonding conditions
NOEUD 606/UZ	- Joint 606 is rigidly bound in Z - axis direction
NOEUDS 102 A 106/RX	)
NOEUDS 201 A 601 PAR 100/RY	) Conditions of symetry
NOEUDS 101/RX, RY	)
CHARGEMENTS	- Loadings
1 PP, TOUT	- First loading case : dead weight. All the results (displacements, stresses, reactions) are to be printed
POIDS PROPRE	
2 DEPLACEMENTS, EFFORTS	- Second loading case : concentrated load (FZ = - 1) applied at joint number 101 Only the displacements and stresses are to be printed
FORCE CONCENTREE	
NOEUD 101/FZ = - 1	
RETOUR	- Return of control of TITUS System.

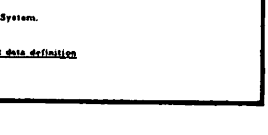


FIG 2 Example of language for input data definition

Fig. 2 - Shows a simple example of using this language for the study of a rectangular plate supported on its 4 corners by columns. An automatic rectangular mesh generation procedure has been used in this case.

2. 1. 3. - Automatic meshing generation

For large problems, above all concerning 3-dimensional structures, an important difficulty arises in the definition of meshing. It then becomes a drudgery to define the geometry, joint by joint and mesh by mesh a procedure of automatic mesh generation is necessary. Faced with the complexity of the problem, and in order to retain maximum service flexibility, we have prepared several procedures of automatic generation, adapted to specific meshing configurations, as well as very general procedures usable for the most complex meshings.

The TITUS system proposes to the user the choice among the following procedures (the word between parentheses indicates the corresponding control)

- individual input of all joints and meshes (GEOMETRIE)
- rectangular meshing whether regular or not (RECTANGLES)
- parallelopipedal meshing whether regular or not (PARALLELEPIPEDES)
- one-dimensional meshing : beams or meridional section of axisymmetrical shells (CONTOUR)
- 1 1/2-dimensional meshing :axisymmetrical shells or translation geometry (SECTEUR)
- any 2-dimensional meshing placed in 3-dimensional space : Thin shells, section of axi-symmetrical or translation structures (STRUCTURE)
- 2 1/2 dimensional meshing : axi-symmetrical and translation structures (STRUCTURE)
- any 3-dimensional meshing : beams, shells, solids (TRIDIMENSIONNEL).

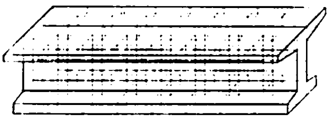


Fig.3

Fig. 3 - Shows an example of space meshing of a T-beam leading to generation of 234 joints and 96 meshes.Structure is described by "macro-meshes". Complete intermediate meshing is generated automatically.

The important problem remains to be able to define simply a meshing relative to geometry of any form. We believe, in fact, that a satisfactory solution resides in a certain "dialogue" between the user and the computer. In this way, the user will define certain parameters of the geometry, in function of which the computer will determine the preliminary coarse meshing (with automatic plotting). Then the user will make the corrections he considers necessary and will enter additional data, in function of which the computer will determine a second, more accurate meshing, and so on . . .

Only such a dialogue, limited to a few exchanges, can rid the user of the tedious tasks while leaving him the possibility of action and decision, necessary in function of the physical nature of the problems.

2. 1. 4. - Remark

These conversational procedures have great influence on the effectiveness of a program. In fact, they determine the user's work proper to simulate his problem and act by means of three factors :

- "easy" or "difficult" aspect of data preparation, influencing the interest of users for the program
- quantity of preliminary work to be supplied
- causes of error.

The last two factors should be reduced to a minimum to the greatest possible extent. As to the more or less easy aspect of data preparation, it should be remarked that it often depends on everyone. Some persons prefer framed printed forms, others a more flexible language (including, however, more risk of errors, through omission basically).

Therefore, faced with the absence of a "universal language" satisfying everybody, the best seemed to be to supply a choice of different procedures, which everyone could use in function of his preferences and of the nature of his problem. The programming principle adopted permits diversification of inputs at will since, in the end, the data are reclassified on a file in a standard manner. Therefore, the way they are inserted does not influence the other procedures of computation and processing a posteriori.

2. 2 - SPECIFIC COMPUTATION PROGRAMS

2. 2. 1 - Theoritical bases

The TITUS system permits design of structures with purely elastic behaviour, subjected only to small strains. For this, the system uses.the classical mathematical theories, viz the Linear Elasticity Theory and its derivatives.

The theory of elasticity is applied for the design of the following structures :

- three-dimensional
- in plane elasticity (condition of plane strains and stresses)
- axi-symmetrical (study in a meridional plane).

Its derivatives mainly consist of :

- the theory of plates and shells for the design of thin structures in bending (their behaviour in their plane concerns plane elasticity).
- the strength of materials for beam design.

These theoretical bases permit tackling the direct design of absolutely any composite structure incorporating one, two or three dimensional elements.

Two remarks should be made concerning the assumptions adopted on "purely elastic behaviour" and on "small strains".

On the one hand, these assumptions define a rather large framework permitting to analyze complex structures, as we have seen above. Besides, these are the hypothesis generally assumed by design departments for computing everyday problems.

Furthermore, the elastic behaviour hypothesis enables us to write linear relations between the stress and strain tensors (Hooke's law). These relations lend themselves well to variational approach and to its processing on a computer. As to the small strain assumption, it leads to a symmetry of these relations (Maxwell-Betti reciprocal theorem), which permits the affirmation of the existence of a state of single structural equilibrium and eliminates certain difficulties in solving linear systems numerically (case of buckling). As to thermic problems, they are studied by means of general thermic equations.

2. 2. 2. - Specific programs

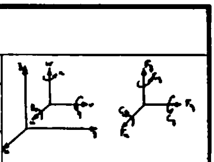
Specific programs make the computations proper to each type of problem. They simulate structures, the behaviour of which is then studied.

These programs constitue an application to programming of adopted calculus theories (variational approach, transfer -matrix, ...). They are also capable of certain general operations such as solving linear systems, calculating support reactions, ...).

The classification of these different programs depends on the following criteria :

- the coordinates used
- the computed displacements
- the allowed loads.

Option	Programs	Coordinates used	Computed displacements	Possible actions	Possibilities
1	Plane elasticity and axi-symmetric structures	x y	u v r	Fx Fy Cz	Plane elasticity and plane frames. Axi-symmetric shells and solids
2	Plates	x y	w p q	Fz Cx Cy	Plates and plane grids
3	Shells	x y z	u v w p q r	Fx Fy Fz Cx Cy Cz	Three-dimensionnal structures, shells and space frames
4	Three-dimensional	x y z	u v w	Fx Fy Fz	Three-dimensional structures, membranes and space trusses
5	Thermic	x y z	T	Q	Thermical conduction in 1,2 & 3 - dimensional structures
6	Thermic in transient state	x y z	T	Q	Thermical conduction in transient state
7	Two-dimensional	x y	u v	Fx Fy	2 - dimensional structures and plane trusses



TERMS AND NOTATIONS USED

- x, y, z : coordinates within the reference axis system (O, x, y, z)
- u, v, w : displ. /to reference axes
- p, q, r : axis rot. / to reference axes
- Fx, Fy, Fz: forces/to reference axes
- Cx, Cy, Cz: axes couples/to reference axes
- Plane frame: plane beam networks loaded in their plane (frames)
- Plane grid: plane beam networks loaded normal to their plane (floors)
- T = temperature
- Q = heat quantity

TABLE I - Overall classification of all specific calculation programs

Table I - shows the classification of the whole set of specific computing programs.

Let us note that options 5 and 6 permit determination of temperature charts usable in the 5 other options for thermoelastic computations.

2. 2. 3 - Types of meshes

The finite element method applies to a very wide choice of discretization and of forms of corresponding meshes. Two criteria are to be taken into account :

- the number of joints (or nodes) associated with a mesh
- the number of degrees of freedom proper to each joint

To our knowledge, we distinguish at present 3 ways in this field :

- the required minimum of degrees of freedom per joint and simple meshes with few joints used in great quantity.
- high number of degrees of freedom per joint and simple meshes with few joints
- few degrees of freedom per joint and complex meshes with large quantity of joints.

Although the second and third ways are promising and being looked into by several foreign teams, we have deliberately chosen the first one, for several reasons :

- its programming is easy
- the computing time of stiffness matrices is short
- these meshes have given satisfaction until now.

It should be noted that the use of such methods in a civil engineering department does not raise the same problems as for an organization of pure research. The engineer is looking more for satisfactory results with good average accuracy, within a minimum of time, rather than for extremely accurate local behaviour. The cost and ease of use criteria matter more for him than the search for academic accuracy.

Table I shows the degrees of freedom (1 to 6) affected to each joint in accordance with the different computing options.

Table II shows the different forms of meshes used by the TITUS system.

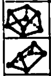
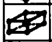
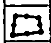
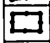
Form	Name	Number of associated joints	Degree of function discretisation	Application
	Bar	2	Linear and cubic	Springs, beams in composite flexure
	Triangle	3	Linear and cubic	)Membranes)Plates)Space shells) -----
	Quadrilateral	4	Cutting into triangles	
	Rectangle	4	2^{nd} and 4^{nd} degree	
	Tetrahedron	4	Linear	)3 - dimensional media
	Hexahedron	8	Cutting into tetrahedrons	
	Prism	6		
	Rectangular parallelopiped	8	2nd degree	
	Triangle	6	2^{nd} degree	)Membranes
	Quadrilateral	8	Cutting into 6 joints-triangles	
	Rectangle	8	3^{rd} degree	
	Triangle	9	3^{rd} degree	

TABLE II - Forms of meshes used by "TITUS' system

Let us note that the discretization of the bars (2 joints meshes) is used only for repartition of loads applied. The beams are in fact, computed directly by the transfert matrices method without performing the approximations of the finite element method.

2. 2. 4 - Limitations

In fact, considering the principle of packed storage adopted (cf. Annex 2), the TITUS system has no a priori limitation on the number of joints, meshes, characteristics, supports, loards ... Only the number of loading cases that it is possible to calculate simultaneously is limited to 10. Any other limitation is due solely to the type of computer used. It is basically a matter of the size of the central memory and of computing accuracy (number of bits per word). The use of a CDC 6600 computer does not seem to raise any problem in this field.

2. 3 - PROCEDURES IN THE OUTPUT END / DOWNSTREAM PROGRAMS

2. 3. 1. - Definition and functions

The limit between the core of the program and the "downstream" processors is poorly defined. Strictly speaking, the core functions are limited to computations of displacements and reactions, without any output. In fact, for practical reasons, it also computes the internal efforts and can, optionally, have the computed results printed out. However, a "downstream" procedure is the normal means of communication available to the user to obtain access to the computation results. This procedure will realize, for the user, auxiliary computations and other long drudgery that is a source of errors . The results are restituted under the form (output device and arrangement) most suitable for the user's requirements. The following are available at the input of these downstream processors :

- data defining the structure and its loads
- for each loading case, results at "computing points"; displacements at the joints, internal efforts in the meshes, reactions at supports. The whole of these data constitue the "structure file".

2. 3. 2 - Requirements to satisfy

2. 3. 2. 1 - What does one wish to be done ?

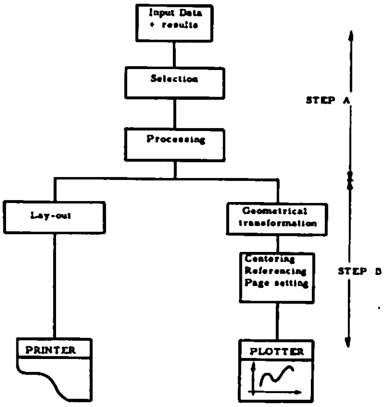


Fig 4 - General diagram of "DOWNSTREAM" processors.

The operations to be realized in a downstream processing are routed as shown on the diagram opposite (fig. 4).

The user must be able to select the components of the results (available in the "structure file" or obtained by prior downstream processing) which he is interested in (displacements, torsional moments, shear-stresses . . .) and the points at which results are desired. The order of these components and of these points is also to be chosen.

On this ordered selection, he must perform posterior processing. Generally, this is done on the basis of elementary loading cases (often without physical reality) of the linear combinations giving the real loads that the structure with stands. In series with these combinations, other processings must be done, either point by point (principal directions and efforts, stresses on extreme fibers . . .) or, on a larger scale (envelopes for a group of combinations, extremes for the selection made, lines of equal value . . .).

The results acquired in this way must be displayed, either on a printer (paging problems)or, above all, on a plotter. Among the displays possible in this case, let us note the variation curves permitting the comparison of values for several loads on the same graph (fig. 20), for several result components (fig. 18), or along the various "routings" through the structure (fig. 17).

Let us also note the charts of principal stresses (fig. 15) or of equal values (fig. 14 and 22) providing a more general display than the curves, but for a single load only. This chart is continuous, for a single component, with equal values, discontinuous for a more characteristic result with the principal stresses. For displaying the displacements the representation of the straines structure is often preferable. All these displays require a geometrical transformation on the "selected points" and, sometimes, on the results associated with them, in order to make the transition from the 3-dimensional space of the structure to representation on the graph (1 or 2 dimensions). Let us note the possibilities of projection on a plane or an axis of development. Concretely, on the plotter, there arise problems of placing, scales, referencing with respect to the structure, graduated axes, title, drawing title block. So, it appears that a downstream processing requires a joint step A, without outputs, and a step B proper to each output component.

2. 3. 2. 2. - How does one wish to achieve this ?

The processing procedures should provide great service flexibility to the user, particularly at the display level. Their structure should be open to permit permanent additions in function of new requirements. They should be easily adaptable to new equipment, for instance cathode ray tube device, which permit direct man-machine dialogue and which it is reasonable to envisage for the inexpensive (with consequent tolerance of errors) downstream processings, in which the rapidity of response is very useful.

2. 3. 3. - "Downstream" processings in the TITUS system

2. 3. 3. 1. - General procedures

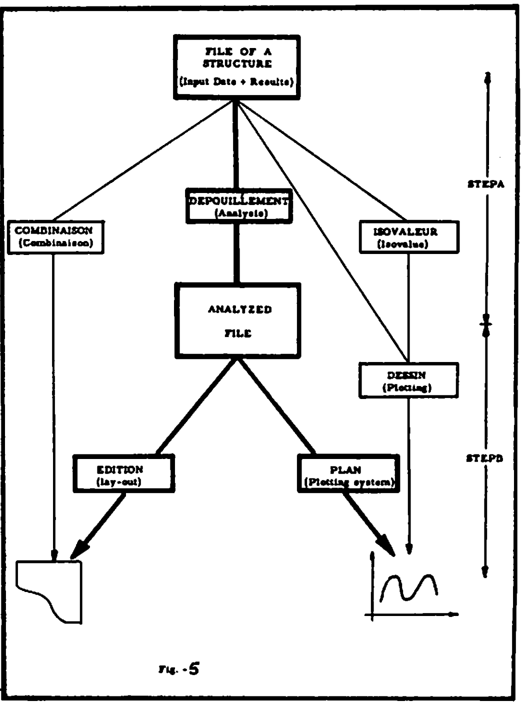


Fig. -5

The central part of the organizational diagram (fig. 5) shows the 2 procedures to be linked in order to perform a downstream processing. "Depouillement" (Analysis, interpretation), performs step A and stores its results in an "analyzed file". Fig. 6, 7 illustrate the analysis by an example, as well as the "Edition" (printing) performed at step B for output on the printer. "Plan" (plotting) displays these results on the plotter. Suitable controls permit :

-positioning of each graph by hierarchized linking of "plans", "cadres" (frame) and "systemes de coordonnées"(system of coordinates).

- definition of the geometric transformation to be made and selection of the chain and the results to be shown under the form of curve or chart of principal stresses (these include all the points of the chain created by "depouillement").

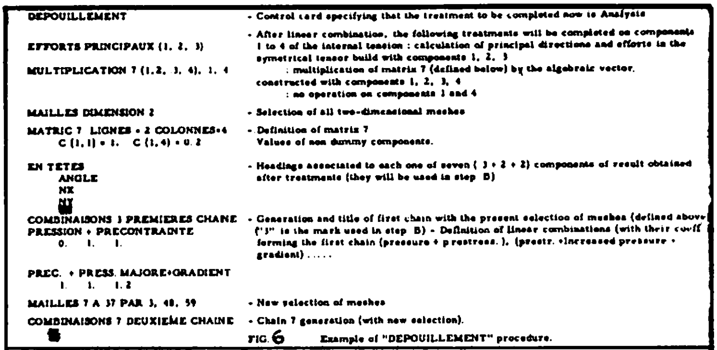
- referencing of the "chain" points in the structure

- trimming the graph : axes, writing, etc . . .

2. 3. 3. 2. - Control languages

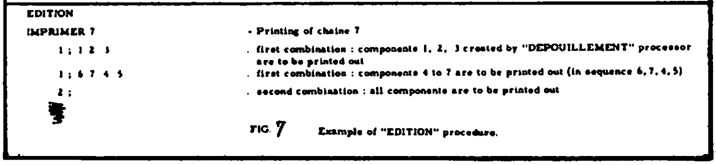
The input languages adapted to each procedure, of which 2 examples are given, must solve two contradictory requirements : flexibility, leaving full latitude to the user, and fast easy use. This dilemma has been solved by placing a choice of means at the disposal of the user:

- a very general language with sharp action
- a choice of several controls, adapted to the different cases, to perform each operation
- macro-instructions, generating a series of sharp statements.
- possibility for "depouillement" (analysis, interpretation) to generate , by adding certain specific controls, the "plan" language, which then perform standard page-placings (fig. 8).



DEPOUILLEMENT	- Control card specifying that the treatment to be completed now is Analysis
EFFORTS PRINCIPAUX (1. 2. 3)	- After linear combination, the following treatments will be completed on components 1 to 4 of the internal tension : calculation of principal directions and efforts in the symetrical tensor build with components 1. 2. 3
MULTIPLICATION 7 (1.2. 3. 4). 1. 4	: multiplication of matrix 7 (defined below) by the algebraic vector. constructed with components 1. 2. 3. 4 : no operation on components 3 and 4
MAILLES DIMENSION 2	- Selection of all two-dimensional meshes
MATRIC 7 LIGNES = 2 COLONNES=4 C (1, 1) = 1. C (1, 4) = 0. 2	- Definition of matrix 7 Values of non dummy components.
EN TETES ANGLE NX NY	- Headings associated to each one of seven (3 + 2 + 2) components of result obtained after treatments (they will be used in step B)
COMBINAISONS 3 PREMIERES CHANE PRESSION + PRECONTRAINTE 0. 1. 1.	- Generation and title of first chain with the present selection of meshes (defined above ("3" is the mark used in step B) - Definition of linear combinations (with their coeff forming the first chain (pressure + p restress.). (prestr. +increased pressure + gradient) . . . . .
PREC. + PRESS. MAJORE+GRADIENT 1. 1. 1. 2	
MAILLES 7 A 37 PAR 3, 48, 59	- New selection of meshes
COMBINAISONS 7 DEUXIEME CHAINE	- Chain 7 generation (with new selection).

FIG. 6 Example of "DEPOUILLEMENT" procedure.



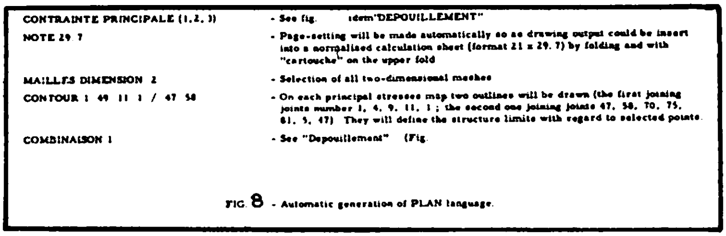
EDITION	
IMPRIMER 7	- Printing of chaine 7
1 ; 1 2 3	. first combination : components 1. 2. 3 created by "DEPOUILLEMENT" processor are to be printed out
1 ; 6 7 4 5	. first combination : components 4 to 7 are to be printed out (in sequence 6, 7. 4. 5)
2 ;	. second combination : all components are to be printed out

FIG. 7 Example of "EDITION" procedure.

2. 3. 3. 3. - Special procedures

To lighten their use, certain procedures "short-circuit" the "analyzed file" (fig. 5).

Let us note "Combinaisons lineaires" (linear combinations), "Dessin" (plotting) for showing strained structures (illustrated by fig. 9), "Isovaleur" (equdvalue) which computes the equal-value lines (on a linear combination of a result) and, which is displayed by "dessin" procedure by one of the types of perspective available.



CONTRAINTE PRINCIPALE (1.2. 3)	- See fig. idem"DEPOUILLEMENT"
NOTE 29. 7	- Page-setting will be made automatically so as drawing output could be insert into a normalised calculation sheet (format 21 x 29. 7) by folding and with "cartouche" on the upper fold
MAILLES DIMENSION 2	- Selection of all two-dimensional meshes
CONTOUR 1 49 11 1 / 47 58	- On each principal stresses map two outlines will be drawn (the first joining joints number 1, 4, 9. 11, 1 ; the second one joining joints 47, 58, 70, 75, 81, 5, 47) They will define the structure limits with regard to selected points.
COMBINAISON 1	- See "Depouillement" (Fig.

FIG. 8 - Automatic generation of PLAN language.

DESSIN	Plotting
SUPPRIMER 1 A 7, 15, 18	Joints 1 to 7, 15, 18 will not appear on the deformed-structure representation
PERSPECTIVE CAVALIERE 45 30	The point of vue will be in the direction 45 degrees, with regard ZOY - plan and 30 degrees above XOY plan - Perspective
DEFORMEE 500 1. 0 1 2	500 times-increased deformation of the structure under combination of loading cases 1 to 3 (with their coeff)
NUMEROTATION 3	Numbering of joints and meshes on the drawing
TITRE 3 27 05 1 PRESSION ET PRECONTRAINTE	Title : position and dimensions of the characters desired

FIG 9 Example of "DESSIN" (procedure) for plotting a deformed-structure

2. 4 - GENERAL PROCESSING PROCEDURE

The general processing procedure consists of an assembly of programs managing all the operations performed. This procedure appears to the user under two aspects :

- control language
- file management.

2. 4. 1. - Control language

In addition to the data specific to each problem, machine processing presupposes the existence of control cards specifying the kind of operations to be made and providing their sequential (linkage).

Fig. 10 describes the general operating of this linkage.

In order to make these control cards clearly explicit, the user has a control language available to him. This language consists basically of keywards, followed, possibly, by 1 or more parameters. The flexibility in using this language is of obvious interest (see Fig. 11).

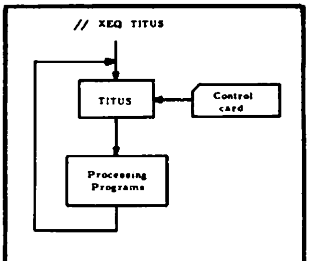


FIG 10 Operational principle of the system

Another advantage of this system resides in the possibility of language extension. It is always possible to add a keyword to those already defined and to make a suitable procedure correspond to it. In this way, any program built subsequently in function of requirements can be entered within the framework of the TITUS system and have assigned to it a corresponding keyword triggering its execution. This is particularly valuable for upstream and downstream programs which can thus be gradually integrated into the system as a whole.

Cards	Significance
DATE	- Date and hour demand
DEFINITION	
Problem data	- Data storage (see § 2-1)
ERREUR VERS 5	- Jump toward the card POINT 5 in case of error
RESOUDRE	- Problem solution (see § 2-7)
ERREUR VERS 5	- Jump toward the card POINT 5 in case of error in solving the problem
CONSERVER EN 103	- Problem (data + results) safeguarding
DESSIN	Plotting (see "DOWNSTREAM" procedures in chapter 3)
DEPOU	- Analysis and combinations of results -(See chapter 3)
PLAN	- Plotting of DEPOU results-(See chapter 3)
ALLER EN 10 POINT 5	- Unconditional jump toward the card POINT 10
EXTRACTIONS POINT 10	- Exceptional extractions to detect the error
HEURE	- Counting the passage time
FIN	- End of processing.

FIG 11 Example of control language

2. 4. 2 - Managing a file

The example given in fig. 11 shows the possibility of keeping the results of a problem in the machine for subsequent processing. These results, associated with the same problem data, constitute the "structure-file" of this problem. The management of a general file ("TITUS file") containing simultaneously several "structure files" (i. e. the data and results of severals problems) constitute one of the basic aspects of the TITUS system imparting to it maximum service flexibility. In fact, it enables the engineer to use again by mean of the control statement "RECHERCHER" - search") the results of a preceding problem at any time, to make any sort of "a posteriori" processings, using the downstream procedures of the system.

The "analyzed file" created at step A of the downstream procedures (see § 2. 3. 3. 1) is contained in the general "TITUS file" (just as the "structure-files"). It benefits in this way by all management procedures of the "TITUS-file" (statements "CONSERVER" = store, "RECHERCHER"= find, "EFFACER" = delete . . .).

This "TITUS-file" (or the "structure files" or the "analyzed files" composing it) may be placed on the most varied supports (disks, drums, tapes, cards) according to the type of computer used and to the desired keeping duration. It is evidently possible to transfer them from one support to another. This leads us to another basic characteristic of this file management : the possibility of mixed processing, i. e. using two computers. The example given (Fig. 12) illustrates the processing of a large problem exceeding the capacity of a small computer (IBM 1130) for instance. It shows that it is nevertheless possible to use the small computer for a maximum of preliminary works and post-processing, the computing part (see § 2. 2)being done on a large computer (CD 6600, Univac 1108, IBM 360/50, 75. . . for instance).

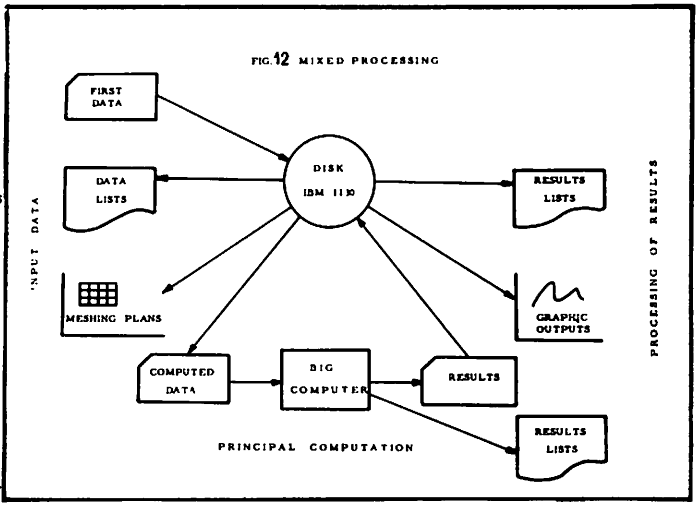


FIG.12 MIXED PROCESSING

This processing method possesses the advantage of reducing the cost while increasing the service rapidity and flexibility.

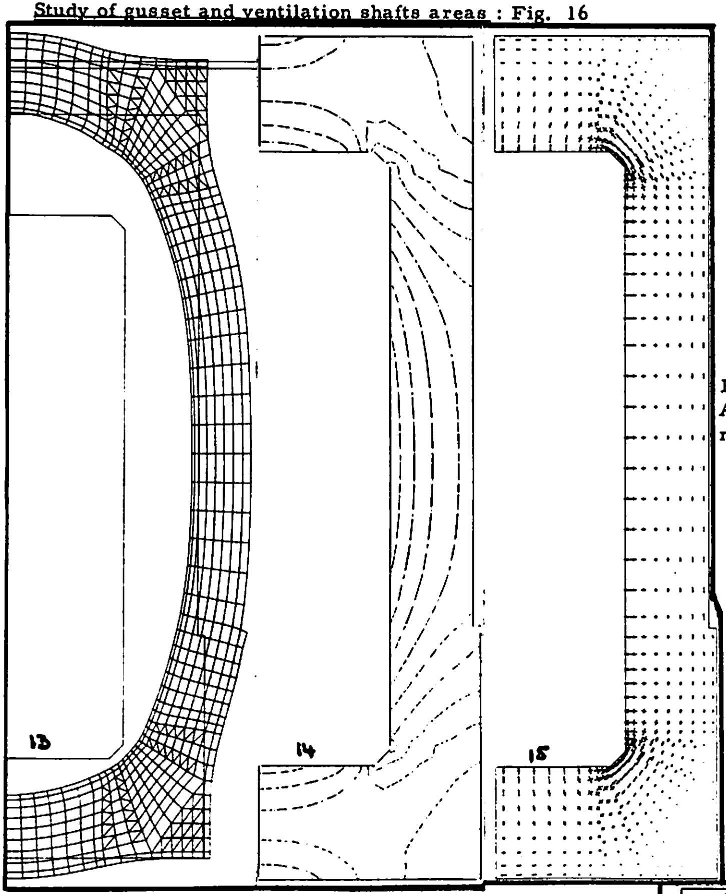
3. EXAMPLES OF STRUCTURES ANALYSED BY TITUS SYSTEM

3. 1 - NUCLEAR REACTOR VESSEL BUGEY I

Study of internal pressure effect : Axisymmetrical structure (incorporating concrete and liner, computed by option 7 - 688 joints, 744 meshes : meshing performed by "STRUCTURE" procedure - Computing time on CD 6600 : Central Processor (64 K) = 151 s
Periph. Processors = 52 s.

Fig. 13 = deformed meshing
Fig. 14 = circonferencial stresses isovalues
Fig. 15 = chart of principal stresses
Fig. 17 = principal stresses // to edge in the lower gusset area
Fig. 18 = RR, ZZ, N stresses along cross-sections.

Study of gusset and ventilation shafts areas : Fig. 16



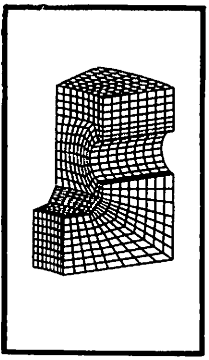
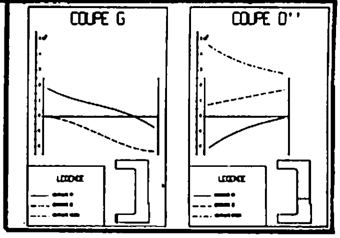


Fig. 16 :
Automatic 3-dimensional meshing.

Fig 18



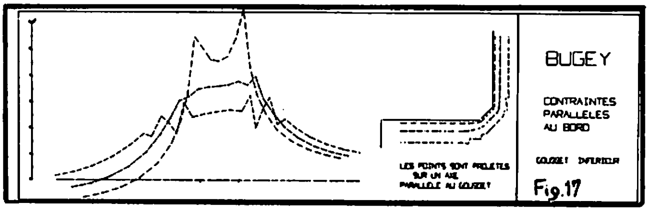


Fig.17

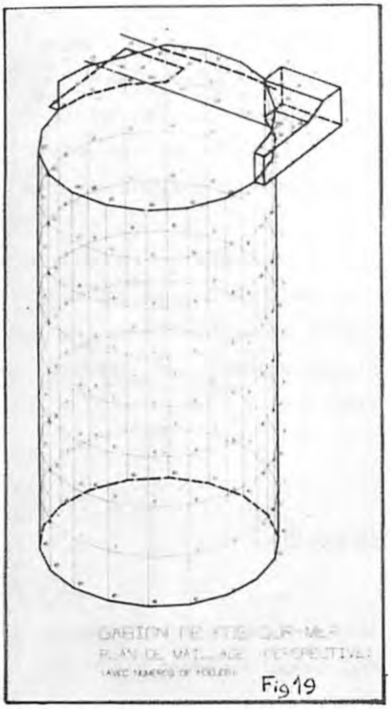


Fig 19

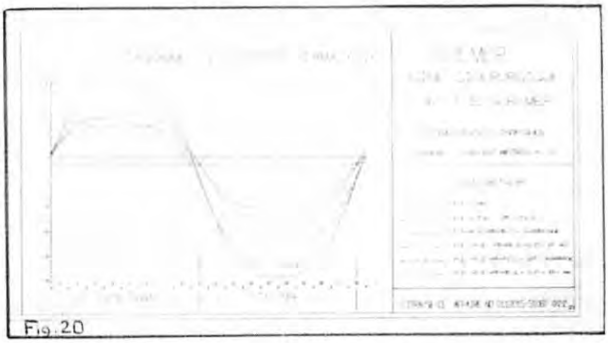


Fig.20

3. 2-FOS-SUR-MER PROJECT - CELL QUAY WALL

Three-dim. structure computed by option 3 incorporating beams (crane path, front beam), shell (cell), slab and block (superstructure)-8 loading cases. 209 joints - 178 meshes - Meshing performed by "SECTEUR" procedure for the cell and by-hand for superstructure. Computing time on CD 6600 : CP (80 K) = 822 s
PP = 277 s.

Fig. 19 : Perspective view on the meshing (with joint numbering)

Fig. 20 : Moment curves under diff. loading cases. The cross-section is along the level of the meshes referenced with respect to the X-axis.

3. 3- COVER OF THE NUCLEAR REACTOR "PHENIX"

Plane structure computed by option 5 (Thermic). Heat exchange coeff. in the whole surface and on the edge - 533 joints - 464 meshes - Computing time :
CP (56 K) = 109s
PP = 25s.

Fig. 21 : Automatic meshing (performed by"STRUCTURE" procedure)of one sixth of the reactor cover.

Fig. 22 : Isotherms

Fig. 23 : Temperature graph represented by a surface.

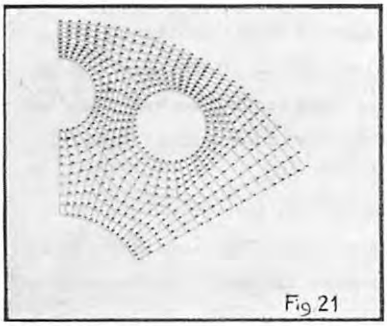


Fig 21

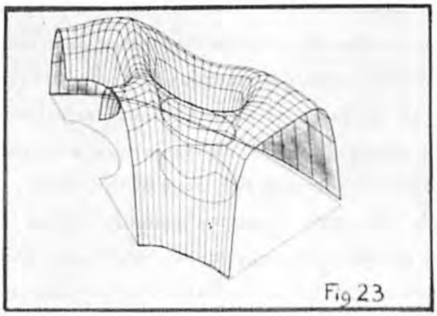


Fig 23

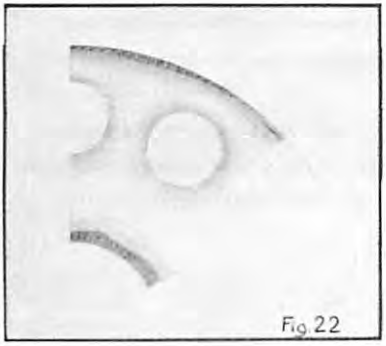


Fig 22

4. CONCLUSION

The finite element method, the mathematical basis of TITUS, has proved its value in recent years. Within the context of practical use in the design departments, we have limited ourselves voluntarily to simple meshes (triangular, rectangular . . .) which, for the time being, provides sufficient accuracy for the user. Thus, we have chosen the way of improving the TITUS system with respect to facilities of use and input-output possibilities, thus evolving parallel with the progress of data processing.
The great flexibility of the system will enable it to be extended as new equipment (terminals, consoles, peripherals, displays . . .) is acquired by the enterprises. It is obvious that any analysis extension will naturally fall in its proper place within the modular structure of the TITUS system. However, in the near future, we are simply trying to improve the present system so as to make the engineer-computer dialogue more effective.

ANNEX I - THEORETICAL FORMULATION

1. ABSTRACT OF THE GENERAL FORMALISM

All the structures analyzed by the system are bound to formally identical equations (linear elasticity). We have then been led to a common general formalism which can be applied particularly to each type of structure being analyzed.

May we recall that the theoretical unknowns of the problem are the value of the displacements at each joint. Writing that the structure is balanced means expressing that the energy spent during its distortion (energy coming from the internal efforts work and from the applied loads) is minimum, under the condition that rigid bonds (punctually applied to joints);are considered. The problem is thus to find the extremum of a function (energy) under conditions.

Computing the energy is achieved in choosing "a priori" the form of the analytical expression of the unknown functions, (displacements) in every mesh. The value (peculiar to each mesh) of coefficients used for this expression is a dependent variable of displacements in joints of this mesh ; this is discreting functions. Here it appears that the chosen approximation will be as much rigorous as the netting will be finer. The elasticity theory (or its subsidiaries) supplies us with the linear relations between stresses and strains (derived from the displacements). Thus the energy will finally be expressed by a quadratic form of the values of the unknown functions at the joints.

To this point, applying a method of the LAGRANGE multipliers allows the expression of the minimum energy under conditions, under the form of the linear equations system, the unknown of which are the values of the joint displacements. Solving this system will supply us with the results (map of joint displacements).

Then applying the linear elasticity relations to these results will give us the stresses, of which we shall compute in fact the mean values in each mesh. The support reactions will be computed by means of the LAGRANGE multipliers.

Note should made that computing one dimension elements (beams) will be made more directly (without discretizing functions) in applying the theory of transfer matrices.

2. CONSTITUTION OF THE LINEAR SYSTEM

By following the method described above, we can examine succinctly how the linear system is constituted. The energy of the structure may be written in its condensed form :

$$E = \frac{1}{2}\ U^{*} .\ A\ .\ U\ -\ U^{*} .\ F \qquad (1)$$

where A = elasticity matrix of the structure (dimensions N. N)
U = displacement vector (dimension N)
F = applied loads vector (dimension N)

N represents here the number of joints of the structure. As to the symbol ($*$) it signifies the transposition of the matrix or vector.

Let us note that the elements of these matrices or vectors are themselves matrices or vectors (dimension 1 to 6 depending on the computing option chosen).

Furthermore, the rigid bonds may be expressed in the form of :

$$B\ .\ \ U\ -\ G\ = O \qquad (2)$$

where B = the rigidity matrix
G = the imposed displacements vector

The Lagrange multipliers method permits writing the minimum of E - see eq. (1) under the condition above - eq (2)

$$E = \frac{1}{2}\ U^{*} .\ A\ .\ U^{*} - U^{*} .\ F + (U^{*} .\ B^{*} - G^{*}) .\ \lambda$$

$$\frac{\partial E}{\partial U} = O \qquad \frac{\partial E}{\partial \lambda} = O$$

where λ = auxiliary vector of the Lagrange unknowns.

A being symmetrical, these expressions become :

$$\frac{\partial E}{\partial U} = A.\ U - F + B^{*} .\ \lambda\ = O \qquad (3)$$

$$\frac{\partial E}{\partial \lambda} = U^{*} .\ B^{*} - G^{*} = O \quad \text{initial conditions} \qquad (4)$$

The completion of bonds to obtain a matrix B of the dimensions N. N permits finally to write the linear system obtained (eq. (3) and eq. (4)) under the following form :

$$\left[\begin{array}{c:c} A & B^{*} \\ \hdashline B & D \end{array}\right] \bullet \left[\begin{array}{c} U \\ \hdashline \lambda \end{array}\right] = \left[\begin{array}{c} F \\ \hdashline G \end{array}\right] \qquad (5)$$

A, the elasticity matrix of the system is symmetrical and, if the joints numbering is done judiciously, inform of a band. In the current cases, one can make B triangular on the left and also in form of a band. D is diagonal and is composed of 1 and O.

ANNEX 2 - PROGRAMMING

Programming such a set of applications cannot be done indepently for each program this being due, for instance, to the parts which are common (routines, data, files . . .). From the start it has therefore been necessary to define a general procedure in which are placed and will be placed the various programs which we have foreseen (and also those which we have not yet foreseen).

All details cannot be explained here. We only wish to mention some of the objectives which have guided us and the corresponding solutions which we have chosen. We must meet the following requirements :

- the programs should be easily converted later on for any scientific computer
- the programs should be easily and progressively written
- the addition of programs meeting new requirements should be easy.

From the programming standpoint, we have chosen a modular form of programming in writing a set of functional routines. These are specialized tools which can be used in any sequence. The fact that they are written in FORTRAN allows the use of them on any scientific computer. This technique is further more and more used at present by the programming teams we know.

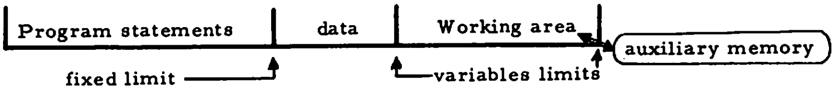
At last, developing a control system (under the control of the monitor of the computer) allowing a sequential interpretation of Control Cards, enables the easy realization and control of the linkage of several operations. This system can be easily transposed on any type of computer.

While the programming of instructions relative to the different modules does not call for any special remarks, data storage and memory management are done less conventionally. The use of large computers in multi-programming (CD 6600, Univac 1108) and of small computers (IBM 1130) has led us to organized a "dynamic" management of the usable central memory. The data defining the problem to be processed are stored in a condensed manner within a "blank common" by means of a calculated addressing system and therefore different from the concept of a predimensioned array in Fortran. This method has two advantages :

- it does not limit, a priori, the size of problems to be processed
- it diminishes the overall dimensions needed by each problem.

The rest of the memory is used as a working area for the various computing procedures, in particular for solving the linear system. If the whole matrix is too large for this area, it is then stored on an auxiliary memory (tape, disk, drum, ECS . . .) and processed by fractions, the size of which is adjusted to the maximum of the effective memory.

This minimizes the access time to the auxiliary memory. So, the organization of the central memory can be presented in the following manner :



Therefore, particularly in multiprogramming, the user can get assigned the size of memory he wishes, in function of the nature of his problem.

ANNEX 3 - NUMERICAL METHODS

The finite element method leads to the formation of large size linear equation systems. The choice of solving methods for such systems is therefore of great importance, both from the point of optimizing the computing time and from that of the accuracy of the results obtained.

After several experiments, we have chosen direct (non iterative) solving methods. Among them we are using :

- direct Gauss method
- Cholevski method
- adapted Gauss method.

All these methods take into account the particularities of the linear system : symmetry, band. In addition, the linear system is solved simultaneously on the different second sides (loads).

1. Gauss method

Its principle is classical :

- a preliminary "descent" triangulates the matrix
- a "rise" gives results in the second sides
- a second"pseudo-descent" reconstitutes the initial matrix and computes the norms and residues of solving.

Let us note that, in fact, during the first descent, the formation of the matrix and its triangulation are performed progressively. This results from the application of a method of "additive inclusion" and of progressive elimination, made possible by suitable internal numbering of the jointsand suitable joints run.

Therefore, the matrix never appears in the central memory under its initial form.

A logic of cutting up and transfer to auxiliary memory permits processing of large size problems. Effectively, the band and symmetrical characters fo the matrix allow to store only a part of this matrix (a triangle on the width) in the central memory.

2. Cholevski method

This method is also classical but differs considerably from the Gauss method. It consists in decomposing the matrix into a product of two triangular matrices , one of which is the transposed of the other in this case of symmetry. The band-character is protected in this operation which is characterized by series of combinations and square roots. This operation can be assimilated to a "descent", after which it is easy to solve the second sides by a "rise".

3. Adapted Gauss method

The classical Gauss method has been adapted by Jensen [5] to the linear systems used in structural computations. Its main characteristics are :

- an internal renumbering of joints to minimize, not the width of the band, but the number of significant non-zero terms of the matrix
- condensed storage of this significant terms
- computations made only on this terms
- obtaining of a "pseudo inverted" matrix permitting to process, a posteriori, any number of loading cases or to make iterations on second sides easily.

Contrary to the previous methods, the concept of band width disappears in this method. The interest of this is obviously enormous, whether when using automatic-meshing procedures leading to great width, or when treating 3-dimensional structures where the matrix is often very wide but relatively "sparse".

In addition, the fact of making triangulations only on the significant non-zero terms reduces the processing time considerably.

Generally speaking, it may be said that all these direct solving methods give excellent numerical accuracy of the results, even for large size systems. The physical significance of the matrices (stiffnesses) and the structural stability simulated by them give a constitution that is particularly well adapted to these methods. According to our experience it is not necessary therefore to use iterative methods often recommended for large systems.

REFERENCES

(1) J. LECOCQ "CALCULS THERMOELASTIQUES dans les CORPS à PAROIS EPAISSES" Industrie Chimique belge T. 32, n° 3, 1967.

(2) ZIENKIEWICZ, "The FINITE ELEMENT METHOD in STRUCTURAL and CONTINUUM MECHANICS", Mac Graw-Hill, 1968

(3) TIMOSHENKO, "THEORY of PLATES and SHELLS, Mc Graw-Hill, 1959

(4) P. LAUNAY, G. CHARPENET and C. VOUILLON, "PRESTRESSED CONCRETE PRESSURE VESSELS : TWO DIMENSIONAL THERMOELASTICITY COMPUTER PROGRAM", Nuclear Engineering and Design (pp 443-459) North-Holland Publishing Company, Amsterdam (1968).

(5) H. G. JENSEN, "EFFICIENT SOLUTIONS for LINEAR MATRIX EQUATIONS", Journal of Structural Division, A S C E (January 1970).