TRANSIENTS BY SUBSTRUCTURING WITH DMAP

Thomas G. Butler
BUTLER ANALYSES

SUMMARY

Automated substructuring in level 16 of NASTRAN was employed as a preface to the solution of a direct transient analysis. The DMAP ALTER statements written to adapt the substructuring for transient purposes are explained. Data recovery was accomplished with transfer functions. Proof of the success of the method is presented with an application to a missile structure.

INTRODUCTION

Substructure analysis capability in NASTRAN has been automated for rigid formats 1, 2, and 3 only. Rather than wait for the extension of automation to rigid format 9, it was found advantageous to use as much of automated substructuring as is now available. Considerable effort was needed to couple the substructure data to the transient analysis and then recover the transient responses of individual substructures.

The two principle features of substructuring that made this effort worth while were the ability to condense small pieces of the matrices at a time and the ability to combine component substructures in different ways. The structure under investigation was liable to damage, and the attraction of substructuring was its ability to substitute damaged components for the hapless predecessors. Many additional advantages of substructuring became evident during the progress of the analysis, but the most notable was the necessity of the analyst to organize thoroughly.

The method presented in this paper could be characterized as a five phased analysis as opposed to the usual three in automated substructuring. Dave Herting of Universal Analytics was extremely helpful in planning the solution path and other substructuring items. Phase one defines the basic substructures without load. Phase two combines components and reduces them to a final pseudostructure but applies deformations for a loading. Phase three recovers influence coefficient matrices for the responses of individual component substructures to the deformations of the pseudostructure. The next phase is the direct transient analysis of the scalar model of the pseudostructure, but is not called phase four. Phase four recovers the response of individual component substructures to the transient excitation using modified static rigid format. A post-processor was written to scan the stress data for the times and locations of parts exceeding a threshold value.

SOLUTION STRATEGY

Answers need to be found to some important questions when one proposes to use substructuring with transients. For instance,

What can and what can't automated substructuring do?
How is the transfer of matrix data from phase two to rigid format 9 accomplished?
What has to be done to get the transient rigid format to recognize the matrices offered to it by substructuring?
How is damping introduced?
How is transient data to be recovered in the component substructures?
How are plots to be obtained?

Outline of Solution

As with any investigation, answers to questions beget more questions. The answers will therefore not be clear-cut until a chain is satisfied. In short, automated substructuring can organize the characteristics of the pseudostructure, as it is modeled for transients, into its stiffness and mass matrices, but will yield neither a damping matrix nor loading. At first blush, it seemed feasible to represent this pseudostructure in scalar form for rigid format 9. Knowing the retained degrees of freedom in the pseudostructure, one then calls for an equivalent number of scalar points for transients. During transients the matrices from phase two substructuring can be introduced by a DMAP ALTER. All matrix generating modules can be by-passed and all matrix partitioning can be circumvented because all SPC's, MPC's, and OMIT's were incorporated during the formation of the pseudostructure stiffness matrix. Taking the response output from the TRD module one can process it by influence coefficient matrices to recover the component responses. This sounds fairly straight forward, so one is encouraged to tangle with the detailed problems.

Data Recovery

Looking first at the data recovery problem using influence coefficent matrices, it helps to think in terms of a super stiffness matrix. In phase two if a unit displacement were imposed in one degree of freedom while holding all others to zero and this is done for each degree of freedom in the pseudostructure, it amounts to an enforced displacement in the form of a unit diagonal matrix. Translating this notion to specifics means that a unit diagonal matrix is needed for substitution as the UGV matrix after module SDR1. Now the SOLVE and RECOVER modules can process the results onto the SOF (Substructure Operating File).

Appendix A contains the details of how these ideas were implemented. The Operations that are important are the DMI input of the unit diagonal matrix, the substitution of the unity matrix for UGV with an EQUIV, the use of a SOLVE command to name the pseudostructure for which the solution chain is intended,

and the use of a RECOVER command to get the solution data out on to the SOF in an orderly fashion so that phase three executions can readily partition the data from the internal substructuring book-keeping scheme.

A rare thing was uncovered that almost shattered this plan. A non-open-endedness, which is quite contrary to the original design philosophy of NASTRAN, was encountered. Only 100 subcases were provided for in any one execution in level 16. But this pseudostructure was of order 916. To impose a unit displacement in each of the 916 degrees of freedom one at a time, meant the assignment of a static subcase for each of the 916 enforced deformations. It was a rare looking Case Control packet that was as large as the Bulk Data deck. In discussing this unhappy event with John McDonough of Computer Sciences, he said that relief of this limitation was already worked out by extending the allowance to 300 subcases. This was small comfort in view of the need for 3 times that amount of relief before this job could execute. The good news was that John had determined, during his investigation preparatory to increasing the allowance to 300, no table restriction or other kind of overflow condition would be confronted if a further extension were attempted. It took two tries to dilate both the standard solution and the automated substructuring section to order 1,000. This has not been generalized yet, but a scheme is believed to be under consideration which will allow the analyst to communicate his needs to the OSCAR and the FIAT by either a DIAG or NASTRAN card entry giving the size of his non-standard subcase array.

The second step in data recovery is to create a set of influence coefficient matrices using automated substructuring phase three. The dimension of the i,j^{th} term of the influence coefficient matrix is

"Displacement in the i^{th} degree of freedom of a Basic S/S per Unit displacement in the j^{th} degree of freedom of the P/S."

where S/S means substructure and P/S means pseudostructure. Any such matrix will designated $[INFL]_{xxx}$, where INFL represents the matrix of terms with dimension u/U=1 and xxx is the subscript to denote the basic S/S by name. This influence coefficient matrix will be used in a post-transient operation to perform the matrix multiplication

$$[INFL]_{xxx}\ [U(t)] = [u(t)]_{xxx},$$

to obtain time varying displacements in a component S/S, where U(t) is a matrix of the P/S response displacement vectors at each of the transient output times, and u(t) is a matrix of the response displacment vectors of component S/S, xxx, at corresponding transient output times.

In computing these influence coefficient matrices during phase three, the number of subcases are required to match the phase two array. This means that in the particular problem, the phase three runs for each component substructure contained 916 subcases. Once again this plan was almost shattered by a bug. For economy purposes the phase three runs were submitted as restarts of the phase one checkpointed runs. Each phase one solution was R. F. 2, and each phase three solution was R. F. 1. The phase three restarts aborted. Only by the chanciest strokes of good fortune did John McDonough happen to have faintly

remembered that somewhere he heard that some difficulty with restarts were overcome by using option 9 on the SOL card, i.e. SOL 1,9. It worked! In effect option 9 avoids the consideration of the optimization features of R. F. 1. This bug appeared during restarts involving change of rigid formats only.

This phase three operation is not out of the woods yet, because a needed data block from the phase one run did not get checkpointed. Appendix B gives the details of how to remedy this defect. Appendix B also explains how the INFL matrix is equated to the UGV matrix and is then written to disc files for the final solution recovery.

Before taking up the problems with transients, the third and last step in data recovery will be completed. Assuming that the matrix of pseudostructure response displacement vectors, U(t), has been successfully written to a disk file, and assuming that the INFL matrices for the several basic substructures have been written to disk files, the task of recovering basic substructure time varying responses in displacements, stresses, and forces is at hand. These jobs will be outside the realm of automated substructuring except that they will be restarted from phase one checkpointed runs. The first task to be performed is to set up the case control such that the vector of response displacements at each output time slice shall be considered as a static solution case. The labelling of each subcase with the output time proves to be a great convenience. Next, the restart data has to be fetched in order to re-establish the internal book-keeping scheme so that the OFP (Output File Processor) module can function in an orderly fashion. Since the product [INFL] [U(t)] will produce a matrix that can be considered the static solution matrix [UGV], no matrix generators or matrix partitioning is needed. The first module needed after the undeformed plot routines is SDR2 so the ALTER packet to bring in the matrices and do the matrix multiply, can start just before SDR2. The outputs from SDR2 are then delivered to OFP to satisfy the output requests in case control. The stress table is output for scanning by a post processor. A normal termination turns control over to the PLOT routine and exits. Details of how these tasks were implemented are given in Appendix C.

Transient Solution

Problems for transient solution begin in phase two of substructuring. Due to a bug in the command "REDUCE" the mass matrix which is produced is designated as square not symmetric. If this were allowed to go uncorrected transient solutions would be 4 times more expensive than expected, because the trailers would telegraph to the DCOMP and FBS modules that unsymmetric routines would have to be called. A scheme was devised, which after much streamlining turns out to be disarmingly simple, using MERGE to change this trailer from square to symmetric. It was no small task to discover how simple it could be to output the pseudostructure stiffness and mass matrices. It was a matter of discovering that there was a module within the substructuring lexicon that was available for addressing explicitly but which was not individually featured. The name of the module is SØFØ. It is incorporated frequently in the listings of DMAP ALTER packets for major commands, so by studying these commands listings it finally registered that if "they" can do it then I can do it. The two complimentary

modules SØFØ and SØFI deserve to be given individual billing in the documentation. Listings of their use for the trailer change to symmetry and for the output of stiffness and mass matrices are contained in Appendix D.

As a preface to the tasks relating to transients, it would be well to ponder what the aims are. At the very least the output result for the pseudostructure should be a matrix of displacement response vectors at a sequence of time slices. Possibly velocity and acceleration outputs may be needed. Plots of the pseudostructure are certainly desireable. It is mandatory that the transients be able to be restarted at a time earlier than the latest time of the preceding run. Restarting the problem with an old set of initial conditions on a new configuration must include acceleration. The model is to be scalar with matrices to be delivered from the phase two pseudostructure. Loads are to be applied to this scalar model. Damping needs to be introduced.

Strange to say, the damping problem will be discussed first. The reason for this order is to settle the question as to whether all matrix generation modules can be by-passed. If uniform structural damping is an acceptable representation of the way the structure behaves, then the damping matrix can be generated by a scalar multiplication of the stiffness matrix, which already exists. This was decided on. Consequently, all matrix generation modules could be by-passed and an ALTER packet could be added which would do the matrix multiplication and whatever related parameter manipulation that would be needed. The Bulk Data of course must contain a PARAM card for the W3 frequency. Myles Hurwitz of David Taylor NSRDC was of inestimable value in helping with parameter manipulations and other system problems.

Now the operation of bringing in the stiffness and mass matrices from phase two is simply a matter of using INPUTT1 and renaming them with an EQUIV statement so that transients can proceed along the normal chain of the rigid format. The Bulk Data of course must contain an SPOINT card containing the number of points equal to the degrees of freedom in the pseudostructure.

The problem of load definition is mainly a matter of book-keeping. Whether the model is geometric or scalar, the nature of the load is such that each component of load has a separate amplification time history so the TABLED1 input data is the same in either case. The DLOAD cards must refer to scalar points, so it is necessary to consult a table to determine which scalar point corresponds to a loaded component of a geometric point. Fortunately, the automated substructuring output items anticipate this need very nicely. The name of the table which tabulates the correspondence between an internal degree of freedom number and the physical point component is "SUMMARY OF PSEUDOSTRUCTURE CONNECTIVITIES." This table is printed in response to the analyst's selection of subcommand "OUTPUT Option 12" during a COMBINE operation. Of course, in this case the table to be used is that associated with the final COMBINE operation which produced the final pseudo-structure. If however, the operation which produced the final pseudostructure configuration were a REDUCE operation, then one needs to consult a pair of tables: the EQSS set of tables and the last PSEUDOSTRUCTURE CONNECTIVITIES table. The EQSS is printed in response to the analyst's selection of option 5 of the subcommand OUTPUT during a REDUCE operation.

The problem of restarting the transient integration at a time earlier than the last of the preceding run is complex. The reason for imposing this requirement is that damage is expected to occur to the structure, but the time of damage won't be known until the results of a previous run are examined. If a stress level is found to be exceeded, the flexibility is provided to substitute a replacement substructure in a damaged configuration. The reconfigured structure will then be restarted at the instant at which the stress level was found to be exceeded. This capability to restart at an earlier time meant that a change in the code of the TRD module had to be made. Simultaneously, a provision in the DMAP listings had to be made to allow for the modification of two book-keeping items. The value of the parameter NCOL had to be set to tell the TRD module at what column in the matrix of the displacements, velocities, and accelerations the data is to be fetched to represent the initial conditions at the time of restart. The table TOL (Transient Output List) has to be enabled so that additions can be made to the table. The changes to the code and to the DMAP were all generously provided by P. R. Pamidi of Computer Sciences.

The problem of initial conditions on a changed configuration could now be handled as a restart at a time specified by the NCOL and a revised TSTEP card without having to go into an external definition using an IC card. Such a restart also provides for initial accelerations to be imposed as well as initial velocities and displacements. Such gymnastics are possible under two conditions. First, the changed configuration must have matrices of the same size (order N), and in the same sequence as the original model. Secondly, the analyst has to be content with the approximate values of the initial conditions on the changed configuration being the same as the terminal values of its predecessor. Under certain conditions this reconfiguring with substructuring allows an analyst to get valuable information about a nonlinear problem using linear analysis.

The final transient task arises because of the particular nature of the UDVT matrix. The CFP arranges a triplet of values for every time. Consequently, UDVT consists of U, $\dot{U}$, & $\ddot{U}$ at t_0, followed by U, $\dot{U}$, & $\ddot{U}$ at t_1, and so forth through time t_N. Only the matrix of time varying displacements is needed for the data recovery phase; therefore the displacements will have to be stripped from UDVT by use of the DMAP utility PARTN. Of course, a partitioning vector will have to be defined by DMI in the Bulk Data to which PARTN can refer. Having the matrix of displacements only, it just remains to use OUTPUT1 to read them onto an external disk file.

Unfortunately, plots of the pseudostructure cannot be obtained because it was defined in R. F. 9 by scalars which have no geometric properties. However, plots of each individual basic substructure can be obtained in the data recovery phase. A listing along with explanations of the ALTER packet for R. F. 9 to implement these tasks are given in Appendix E.

APPLICATION

The technique of combining automated substructuring with direct transients was applied to the analysis of a current missile. Figure 1 shows a picture of the hardware. Figures 2 through 7 depict the undeformed plots of the substruc-

tures. Figure 8 shows a chart giving the evolution of the analysis from the definition of individual substructures, the combining into pseudostructures, through transient analysis and finally data recovery. This chart is usually called a substructure analysis tree or simply a tree. In this case, however, there are five phases instead of three. Figure 9 displays the variations in the tree under different configurations with a minimal of annotation.

The technique proved to be quite successful in that the analytical results compared quite favorably with two full scale tests.

HINDSIGHT

Naturally, when one achieves a certain measure of success with a task, the thinking process does not end abruptly. It has occured to me that some things could have been done differently. One is in the area of dynamic loads. Allow me to retrogress for a moment. During the analysis great care was taken to control the band and the density of the K matrix. Banding was performed on each substructure. Condensations were performed on matrices whenever possible when they were of small order; e.g. OMIT's were introduced in phase 1 as deeply as possible without interfering with connections or loads; reducing was used in phase two at connection interfaces without interfering with loads. Condensations, after loads were defined in transients, were considered but abandoned, because the pay-off among trade-offs was not immediately evident. Parallel condensations would have to be performed in phase two and in R. F. 9, because matrices would have to be delivered to R. F. 9 in uncondensed form, then a phase two condensation would have to follow the matrix transfer to match the condensation that would take place in transients, because data recovery of necessity is based upon the matrices in "SDISPLACEMENT" form from transients. This penalty of double condensation plus the increased density may outweigh the advantage of order reduction in transient DCOMP and FBS. A way of omitting the double condensation penalty has surfaced after the analysis. The DAREA loading could have been represented as "unity" static loading in phases one and two with condensations including some of the loaded points. The "static load matrix" could be examined after phase two and before transients to arrive at the weighting functions different than 1 for the DAREA cards. Time amplifications would have to be adjusted accordingly.

Another post hoc idea occurred in the area of plotting. After the transient analysis of the scalar model was complete, another grid point model could be assembled with a grid point for every scalar point wherein the other 5 degrees of freedom would be constrained. An ALTER packet to admit the static and deformed PLOT modules and substitute the scalar UDVT for the "SDISPLACEMENT" input to the VDR module would allow the plots to be obtained for the pseudostructure transient response at a small cost.

CONCLUDING REMARKS

A technique for combining automated substructure analysis with transient analysis has been devised and successfully applied. The technique can be summarized in the form of a recipe to assist an analyst, who may want to try this technique, as to what factors have to be taken into account.

Recipe

1. Run S/S through Ø2. Combine & reduce to produce the transient P/S model.
2. Run Ø2 SOLVE & RECOVER operations with ALTER packet to read [I] matrix into UGV data block and supply subcases for each of the N degrees of freedom.$
3. Run Ø2 to change MMTX from sq. to sym. using MERGE* packet. Read KMTX & MMTX onto external disk file using SØFØ command and OUTPUT1 packet.
4. Set up transients as a scalar problem. Read KMTX & MMTX in from external file. Supply loading data to scalar degrees of freedom.
5. Apply ALTER packet to jump around matrix generators, to build damping matrix, to set transient parameters, to partition displacements from UDVT, and to provide for restart @ earlier time than last.**
6. Run Ø3 as SØL 1,9* with ALTER packet to correct SDR1* and output INFL.
7. Run Ø4 as SØL 1,9* with ALTER packet to jump around matrix generators, to multiply for basic S/S transient displacement response, and output stress table.

$ N is limited to 300 in standard level 17.
* These steps will not be necessary in an operating version with these bugs corrected.
** Level 17 has this feature incorporated in the code.

Figure 1
MISSILE HARDWARE

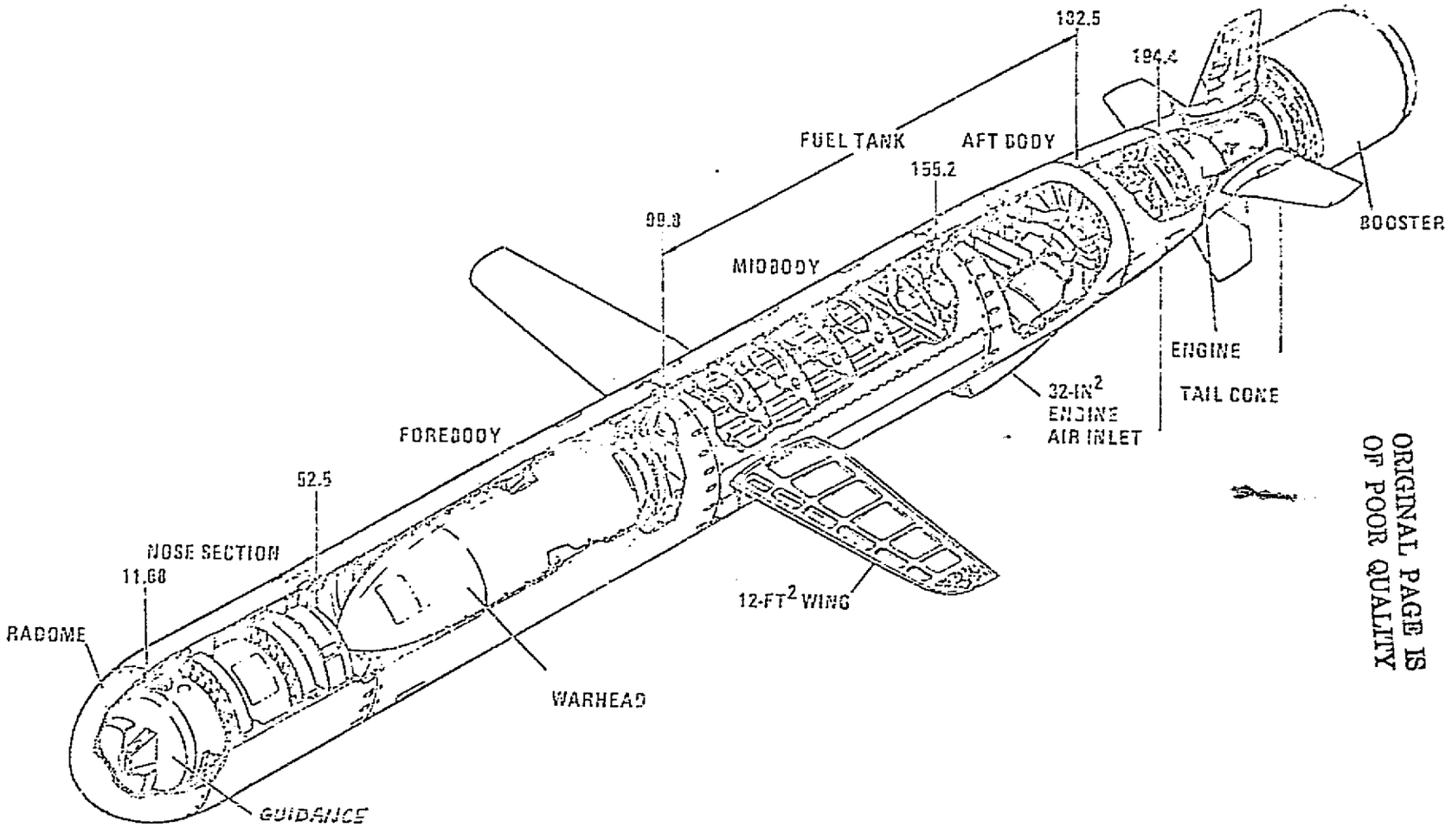
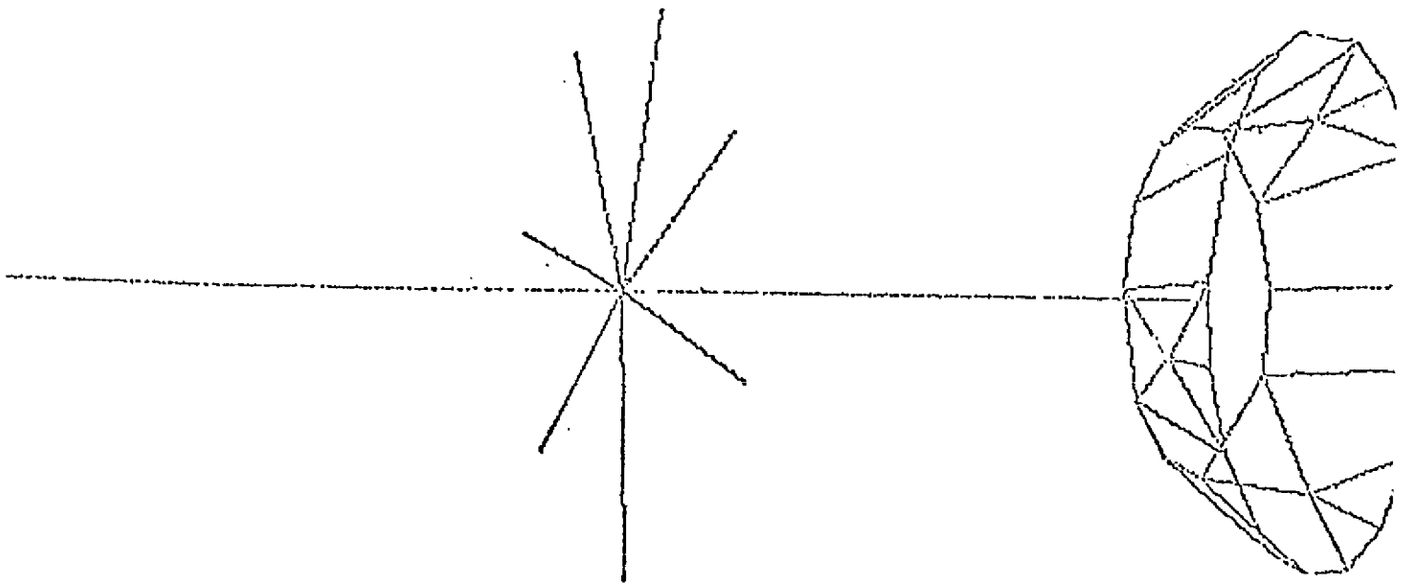
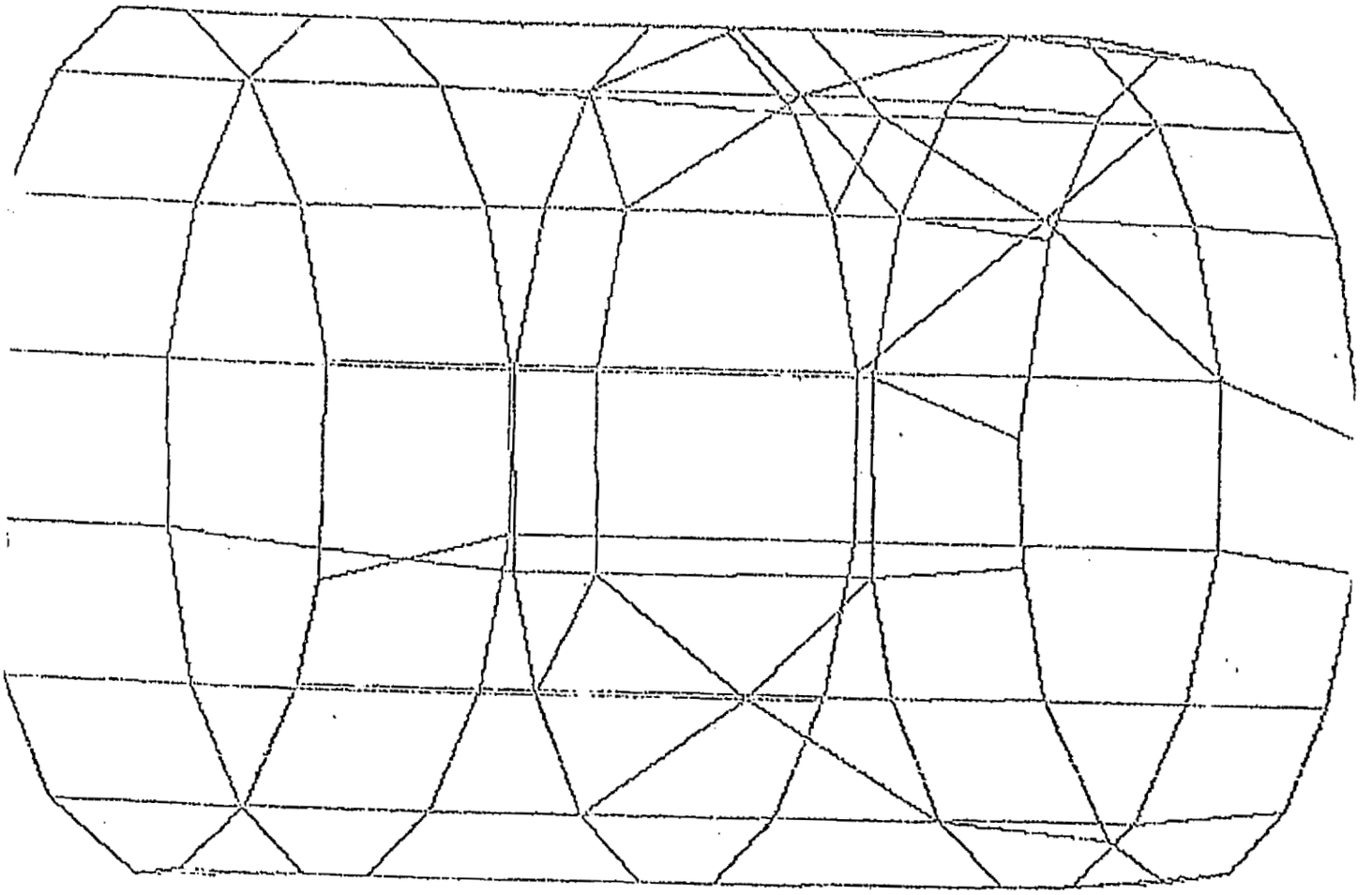


Figure 2



BUL

Figure 3



SHL

Figure 4

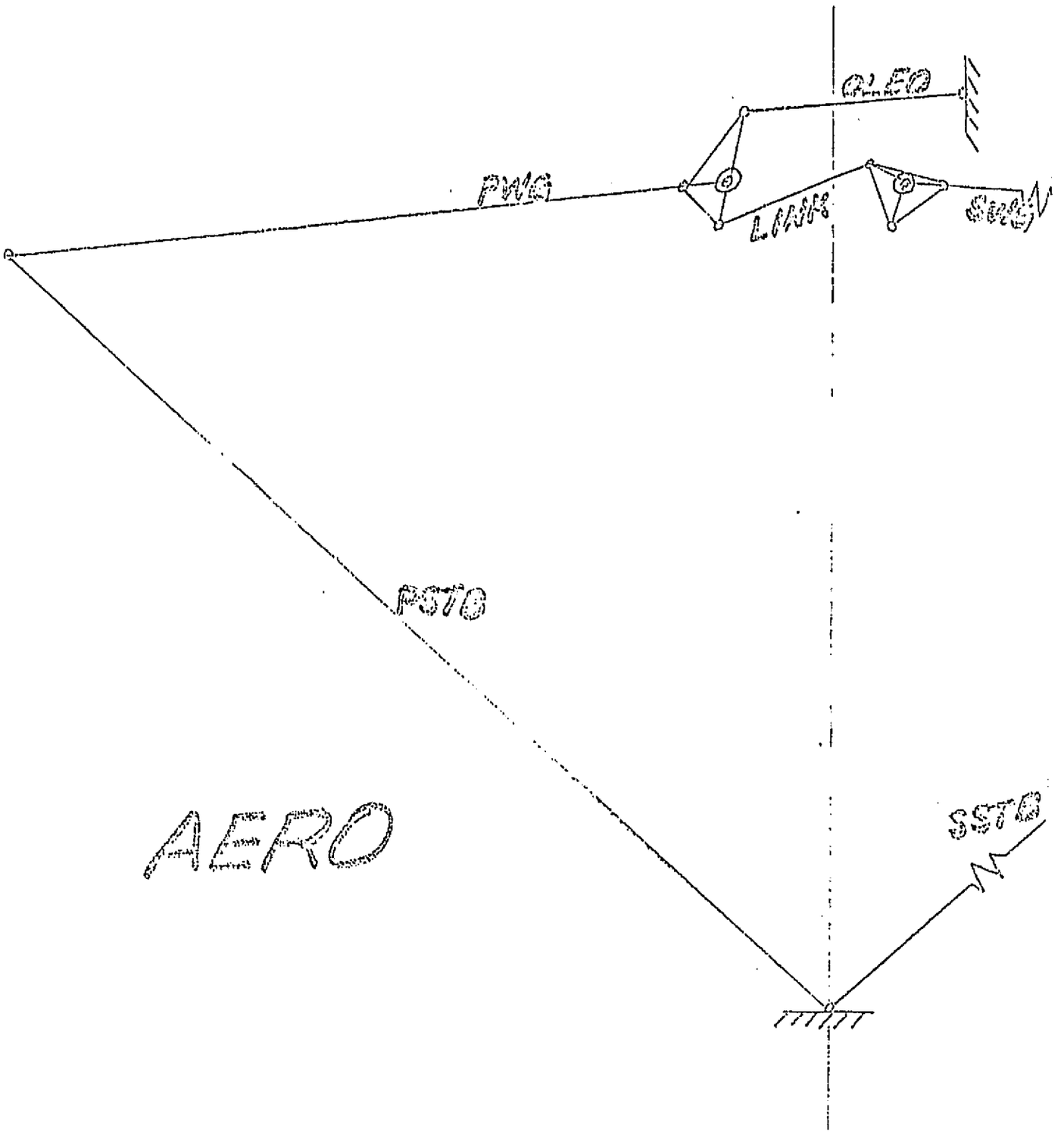


Figure 5

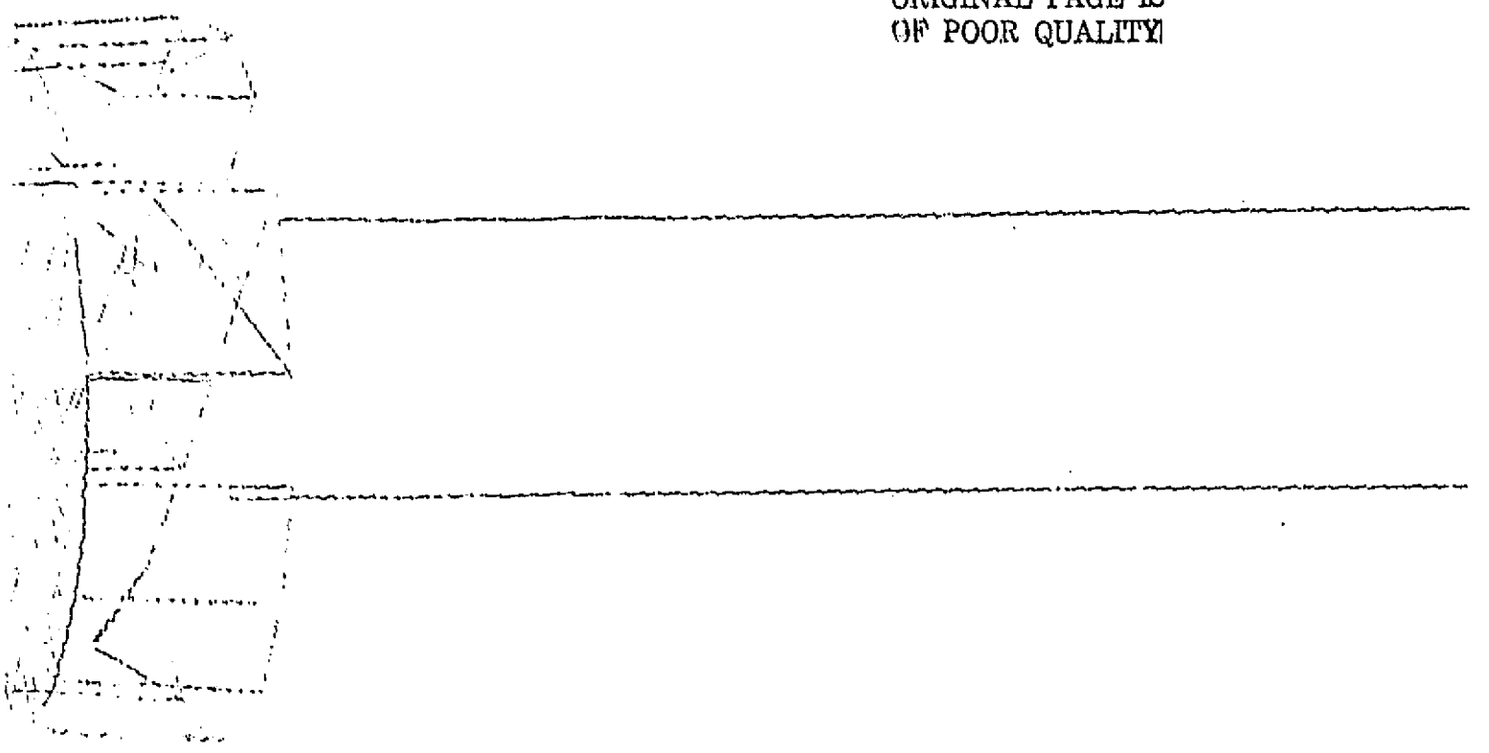


Figure 6

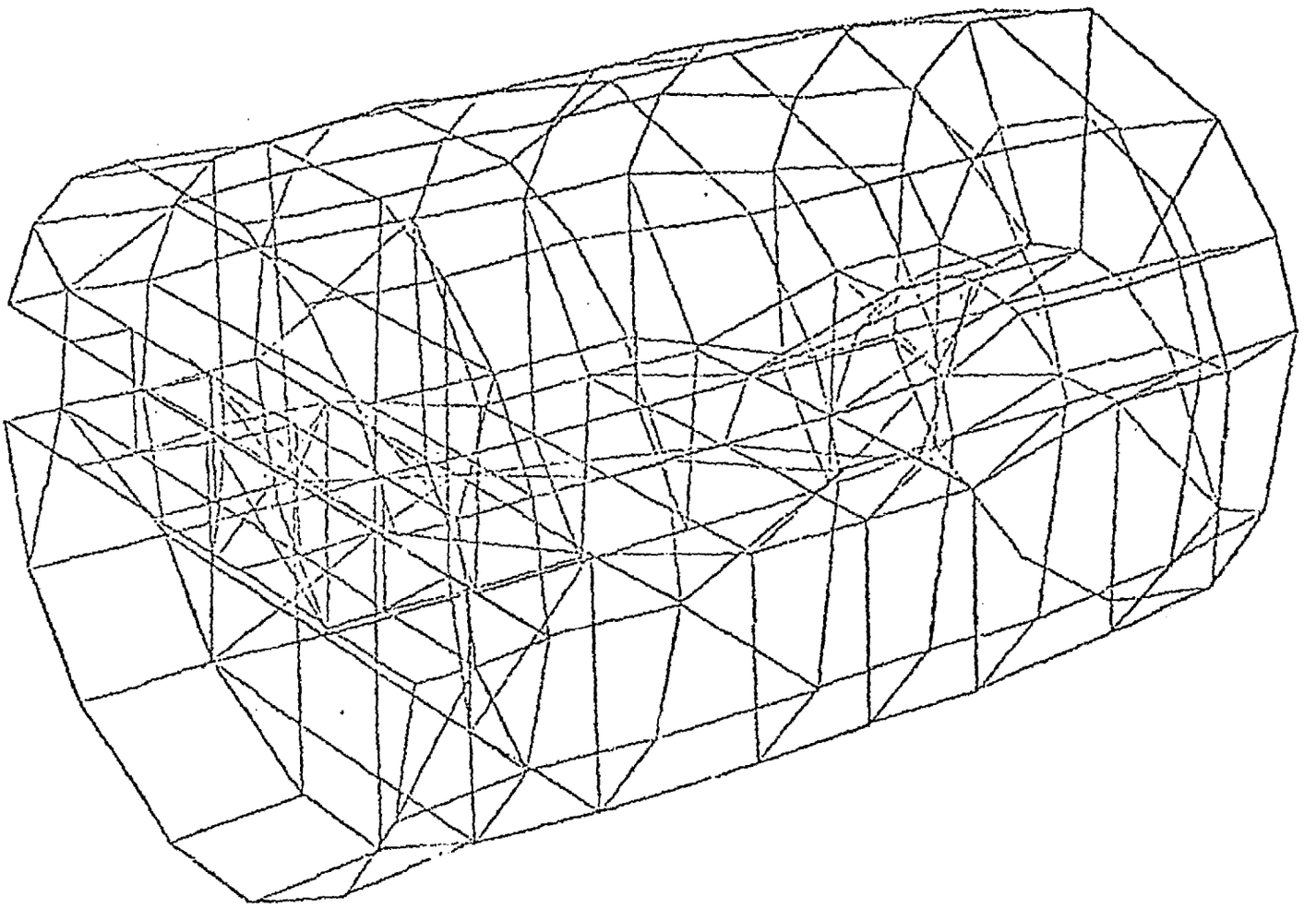
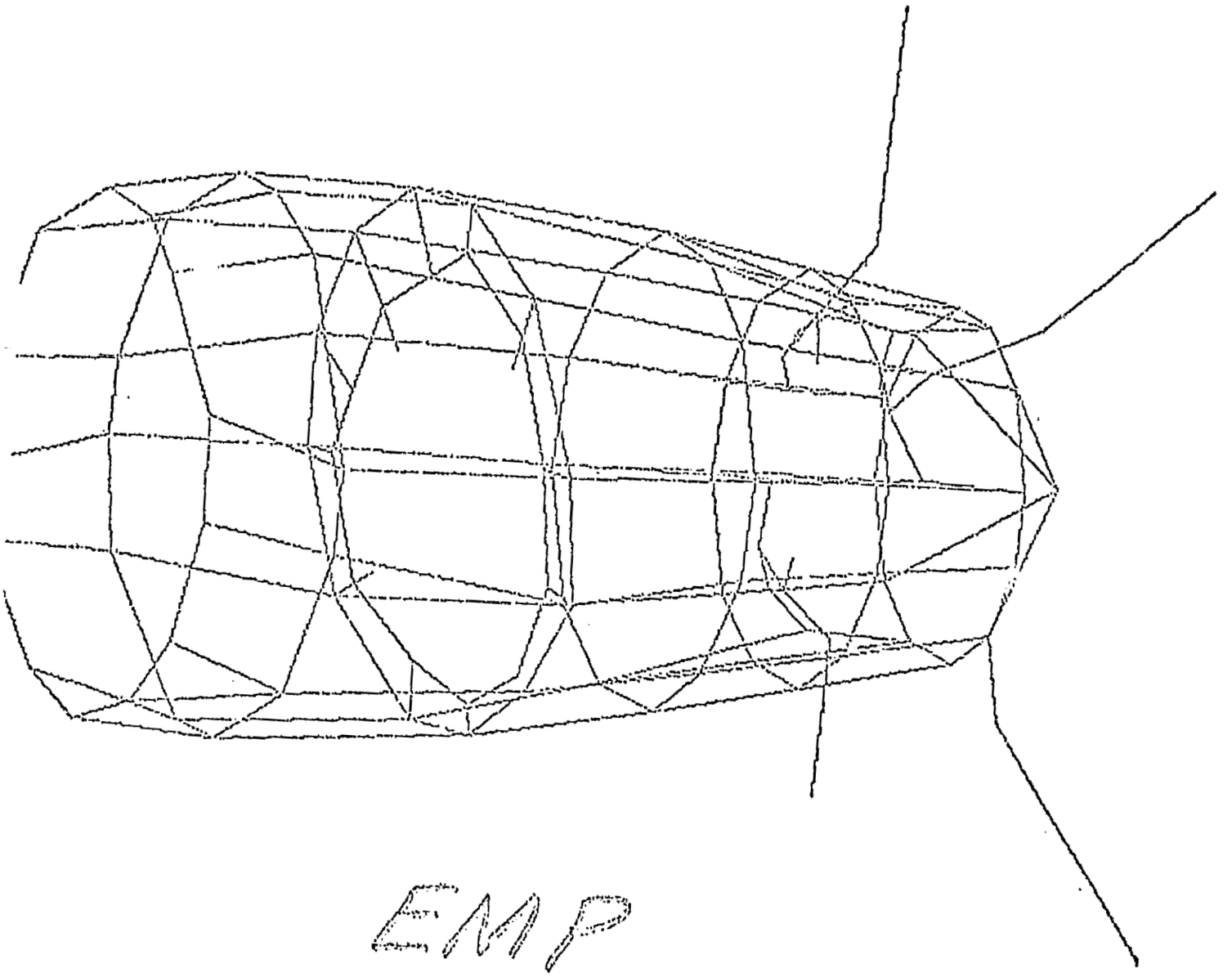


Figure 7



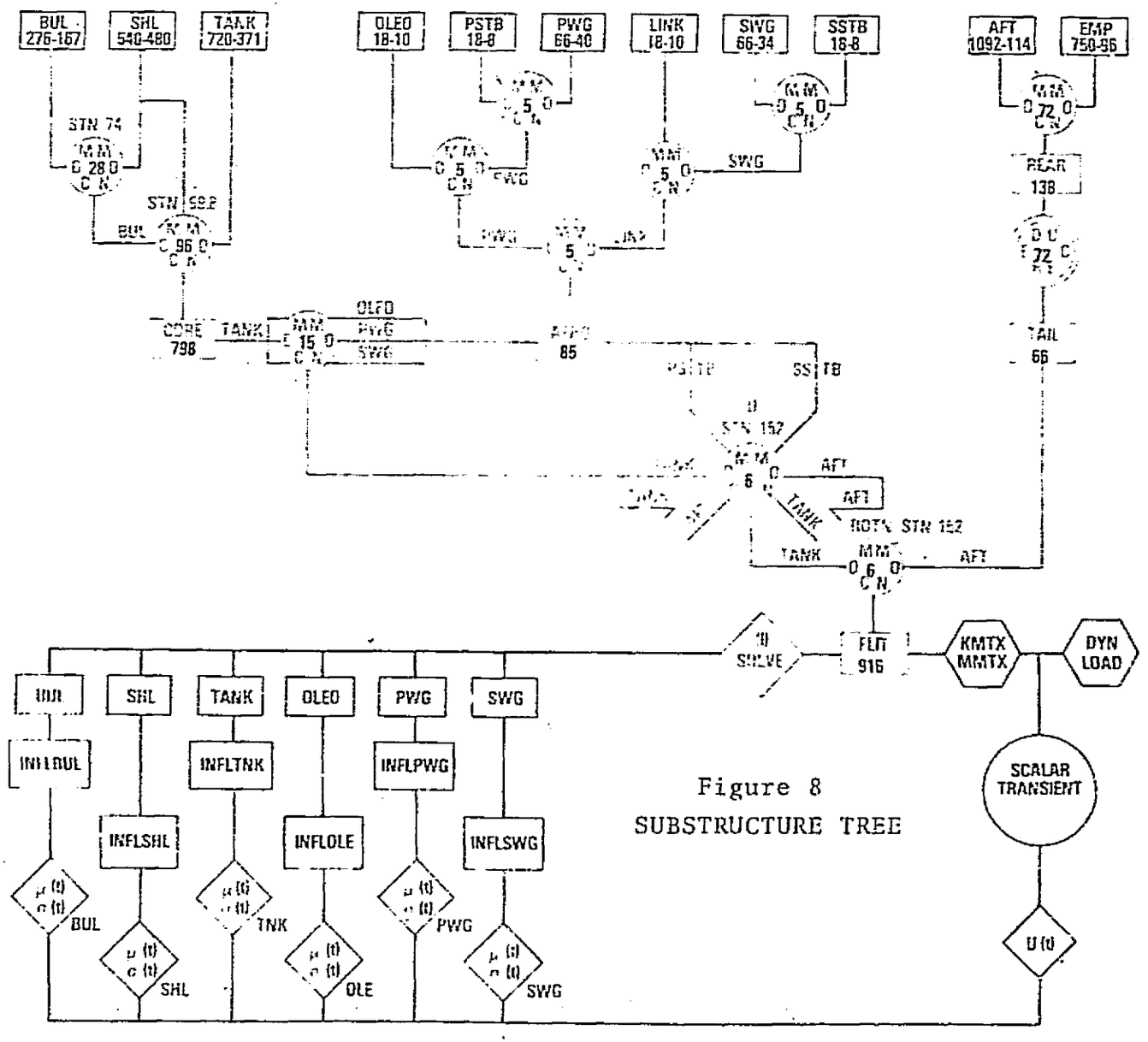
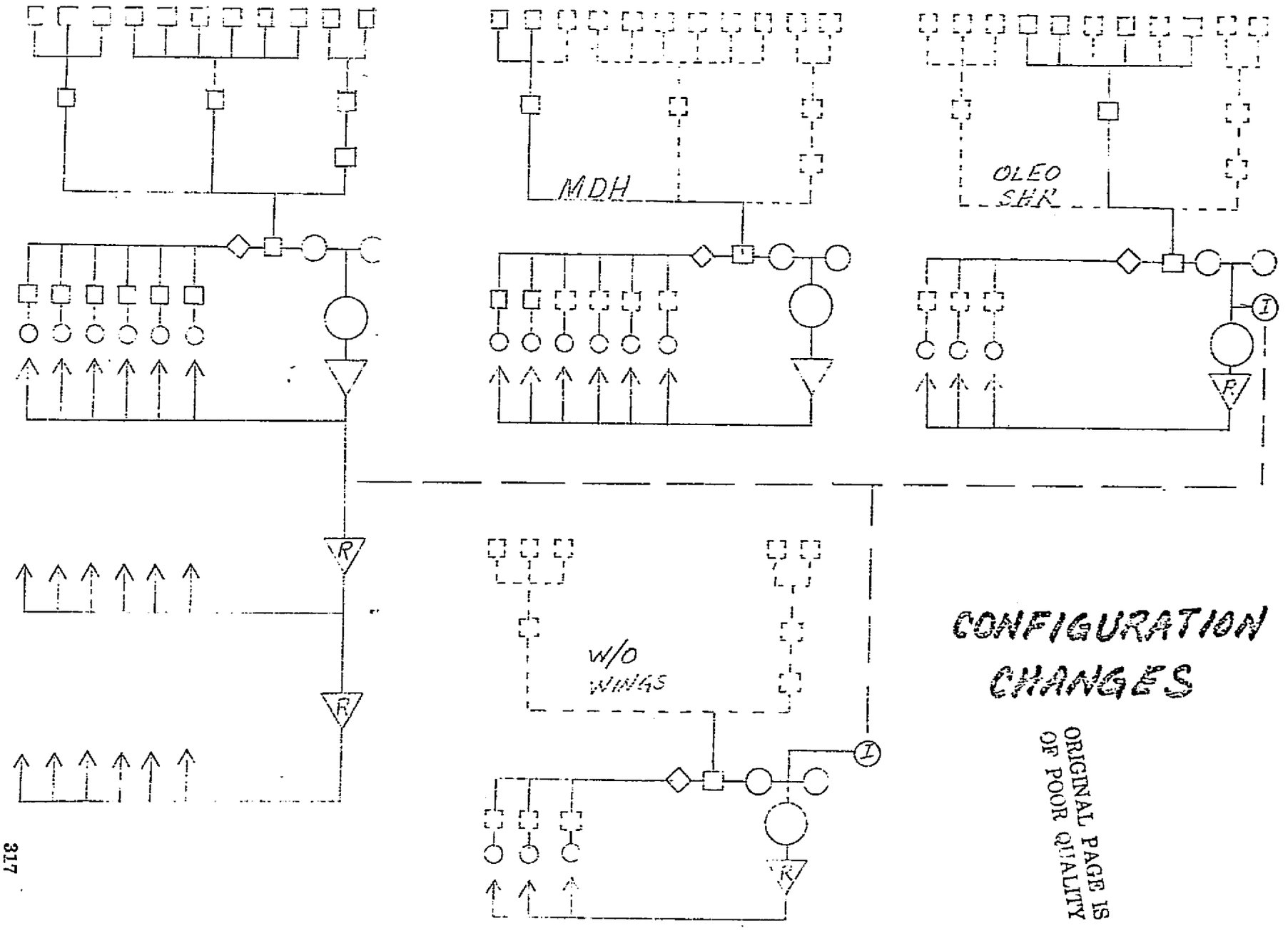


Figure 8
SUBSTRUCTURE TREE

Figure 9



APPENDIX A

DMAP ALTER TO R.F.I. FOR UNITARY MATRIX LOADING WITH S/S SOLVE

This appendix supplements the description of SUBSTRUCTURE Phase II Solution Strategy. The Phase II DMAP ALTER statements are given first, followed by explanations according to statement number.

```
1.  ALTER 100
2.  JUMP TGB $
3.  ALTER 126
4.  LABEL TGB $
5.  EQUIV PARTUGV, UGV/ALWAYS $
6.  ENDALTER
```

Statement Explanations:

1. Since these ALTER statements are used in conjunction with S/S commands SOLVE and RECOVER, this ALTER must not interfere with the automated ALTER's associated with these commands, i.e. these must be avoided 2, 4 to 5, 9 to 22, 29 to 30, 41, 58 to 61, 73 to 78, and 134 to 164. The ordinary preparation of matrices such as constraints and omits will in general be needed. The operations to be eschewed as unnecessary and costly are decompositions, load assembly, FBS (Forward Backward Substitution), and SDR1 (Stress Data Recovery). Begin the ALTER at statement 100 in R.F.1 because constraints and omits have been completed by this point, and it precedes the decomposition.

2,3,4. Start the avoidance of unwanted operations with a JUMP command to a label located after the recovery of UGV in module SDR1.

5. The name o the unitary matrix input via DMI cards is PARTUGV. Since it is desired to have NASTRAN recognize this unitary matrix as the displacement solution, it must be labeled UGV for succeeding modules to so recognize it. There PARTUGV is equivalent to UGV. The parameter ALWAYS was used to make this EQUIV "take" because it was established earlier in the automatic ALTER's with a value of negative one.

APPENDIX B

DMAP ALTER FOR CORRECTING SDR1 AND OUTPUTTING INFLUENCE COEFFICIENT MATRIX

This appendix supplements the description of SUBSTRUCTURE Phase III Solution Strategy. The Phase III DMAP ALTER statements are given first, followed by explanations according to statement number.

```
1.  ALTER 126, 126
2.  COND      NOUOOV, OMIT $
3.  FBS       LOO,,PO/UODK/C,N,1/C,N,1/C,N,1/C,N,1 $
4.  CHKPNT    UODK $
5.  EQUIV     UODK, UOOV/ALWAYS $
6.  CHKPNT    UOOV $
7.  LABEL     NOUOOV $
8.  SDR1      USET, PG, ULV, UOOV, YS, GO, GM, PS, KFS,
              KSS,/UGV, PGG, QG/V,N,NSKIP/C,N, STATICS $
9.  OUTPUT 1, ,,,,//C,N,-1/C,N,3 $
10. OUTPUT 1 UGV,,,,//C,N/O/C,N,3 $
11. EXIT $
12. ENDALTER
```

Statement Explanations:

1. This is a restart of a SUBSTRUCTURE Phase 1 checkpoint. A required input data block UOOV for module SDR1 did not get checkpointed in the Phase I run because SSG3 was automatically ALTERed out, so therefore is not available for restart and its absence would cause an abort because UOOV is not allowed to be absent as an input data block if an OMIT occurred.

Since the SDR1 module in R.F. 1 is statement number 126, the patch for remedying this defect must be inserted either ahead of or in place of 126. In this case it is inserted in place of 126, because constraint forces QR were not checkpointed in the Phase I run either, necessitating the removal of QR as an input data block to SDR1.

2 thru 7. These six statements are intended to provide for both the presence and absence of a Phase I OMIT. If there were a Phase I OMIT, the parameter named OMIT would be given a positive value when output from GP4 and the COND conditional jump to label NOUOOV would not take; thus causing the operation sequence to pass to statement 3, FBS. If there were no Phase I OMIT, the parameter named OMIT would have retained its negative one default value, and the COND conditional jump to label NOUOOV would "take" and the FBS module would be by-passed.

The reason that it is desired to cause this conditional jump is that UOOV would have existed albeit as a purged data block, and would satisfy the input requirements of SDR1. The EQUIV of data block UODK to UOOV uses parameter ALWAYS because ALWAYS was defined to be negative one by automated substructuring. Data blocks UODK and UOOV are checkpointed before and after the EQUIV because the DMAP compiler is very particular about tidiness in specifying the precedents and antecedents in EQUIV.

3. Data block PO is required to be supplied by the analyst as DMI input. It is of order "A" rows and "0" columns and is null. This will create a null UOOV. The reason that UOOV

should be null is that no load has been put on any omitted points.

8. The call statement for SDR1 must be written without an input data block appearing after KSS. It is not necessary to remove output data block QG even though QR is absent, because SDR1 will generate QG as purged in the absence of QR.

Output data block UGV will now be the basic substructure influence coefficient matrix.

9,10. These modules will output the influence coefficients to a NASTRAN file INP3. If the JCL is written to make a disk file of INP3 it should be named INFLxxx for ready identification.

11. An exit is taken after outputting the INFLxxx data because no more processing is needed.

Statements 4, 5, 6 could have been omitted and UODK could have been used in place of UOOV as an input data block to SDR 1, because USET would have sensed that UOOV was purged during execution of DMAP #71 of R,F, 1 in the event of no OMIT, and would have relieved SDR1 from requiring UOOV as input to SDR1.

APPENDIX C

DMAP ALTER TO R.F.1 FOR POST-TRANSIENT DATA RECOVERY

This appendix supplements the description of STATIC DATA RECOVERY PHASE IV. The Phase IV DMAP ALTER statements are given first, followed by explanations according to statement number.

```
1.  ALTER       39, 155
2.  INPUTT1     /,,,,/C,N,-1/C,N,3 $
3.  INPUTT1     /INFLxxx,,,,/C,N,O,/C,N,3 $
4.  INPUTT1     /,,,,/C,N,-1/C,N,4 $
5.  INPUTT1     /FLIT#U,,,,/C,N,O/C,N,4 $
6.  MPYAD       INFLxxx, FLIT#U,/TRANxxx/C,N,)/C,N,1/
                C,N,O/C,N,1 $
7.  PARAM       //C,N,MPY/V,N,ALWAYS/C,N,-1/C,N,L $
8.  EQUIV       TRANxxx, UGV/ALWAYS $
9.  CHKPNT      UGV $
10. SDR2        CASECC, CSTM, MPT, DIT, EQEXIN, SIL,
                GPTT, EDT, BGPDT,,,UGV, EST,,/,,OUGVI
                OES1, OEF1, PUGV1/C,N,STATICS/
                V,N,NOSORT2=-1 $
11. SAVE        NOSORT2 $
12. OFP         OUGV1,,,OEF1, OES1, //V,N, CARDNO $
13. SAVE        CARDNO $
14. OUTPUT2,    ,,,,//C,N,-1/C,N,11 $
15. OUTPUT2     OES1,,,,//C,N,O/C,N,11 $
16. OUTPUT2,    ,,,,//C,N,-9/C,N,11 $
17. ENDALTER
```

1. The operations that are needed in this step are fetching of restart data, forming of basic time varying displacement matrix, exercising of SDR2 and OFP, plotting of deformed responses, and outputting of stress data to the post-processor. Call statements for SDR2 and OFP in standard form process more than is needed here, hence their input and output data blocks are destined to be rewritten consequently everything from before matrix generators to just before structure plots can be by-passed. Hence, the ALTER control causes a jump from statement 39 to statement 155.

2,3,4, and 5. Influence coefficient matrix [INFLxxx] is read in from disc file via INP3. Transient response FLIT#U is read in from disc file via INP4.

6. The matrix multiply operation [INFLxxx] x [FLIT#U] = [TRANxxx] gives the Basic Substructure time varying response TRANxxx.

7,8, and 9. A control parameter ALWAYS is assigned the value -1 to make the succeeding EQUIV operation "take" so that the Basic Substructure time varying response TRANxxx is equivalenced to the static solution matrix UGV. Then UGV is checkpointed.

10 and 11. Basic Substructure time varying displacements, element forces and element stresses are recovered according to the specifications stated in Case Control. The output parameter NOSORT2 is saved.

12 and 13. The displacements, forces, and stresses from SDR2 are formatted for printing. The output parameter CARDNO

is saved.

14, 15 and 16. The table of stresses OES1 are output to a FORTRAN file and stored on a disc with dataset name STRSxxx. This is now in a form that the post processor scan program can access it.

APPENDIX D

MMTX TRAILER CHANGE AND MATRIX TRANSFER FROM SOF TO EXTERNAL

This appendix supplements the description of TRANSIENT SOLUTION STRATEGY. The DMAP listing is given first, followed by an image of the partitioning vector, then the explanations according to statement number.

```
1.   BEGIN  $
2.   SOFI       /K1,M1,,,/C,N,+1/C,N,MFLT/C,N,KMTX/C,N,MMTX  $
3.   MERGE      M1,,,,BLNKVEC,/SQRSYM/C,N,2/C,N,1/C,N,6  $
4.   PARAM      //C,N,ADD/V,N,TRALR/C,N,0/C,N,-1 $
5.   EQUIV      SQRSYM,M2/TRALR $
6.   SOFUT      //C,N,+1/C,N,MFLT/C,N,EDIT/C,N,2/V,N,ZXX/V,N,ZXX/
                C,N,DUMPARAM/V,N,ZXX/V,N,ZXX/V,N,ZXX/V,N,ZXX/
                V,N,ZXX $
7.   SØFØ       ,M2,,,,//C,N,+1/C,N,MFLT/C,N,MMTX $
8.   SOFUT      //C,N,+1/C,N,TOC/C,N,SOFP/C,N,0/V,N,ZXX/V,N,ZXX/
                C,N,DUMPARAM/V,N,ZXX/V,N,ZXX/V,N,ZXX/V,N,ZXX/
                V,N,ZXX $
9.   OUTPUT1,   ,,,,//C,N,-1/C,N,0 $
10.  OUTPUT1    K1,M2,,,//C,N,0/C,N,0 $
11.  END        $

DMI  BLNKVEC  0    2    1    1        916    1    +PART
DMI  BLNKVEC  1    1
```

2. The utility SOFI brings information form the SOF into the current execution. The name of the P/S is MFLT and the two data blocks belonging to MFLT, KMTX and MMTX are being requested and being given temporary temporary data set names K1 and M1 respectively.

3. A merge operation of the mass matrix M1 with a null matrix is performed by using a 916th order null partitioning vector. The partitioning vector is easy to specify as shown on the example DMI card. Ablank entry is given for the first row of the first column and implied blanks for all others. The output matrix is named SQRSYM. The 3rd parameter states that the trailer of the output matrix is to be symmetric.

4,5. A parameter is defined for use in the EQUIV statement so that the output mass matrix can conform to the substructuring format for naming.

6,7,8. SOFUT is used to purge the old MFLT square mass matrix M1 from the SOF. SOFO is used to output the summetric mass matrix M2 from the current execution to the SOF record of P/S MFLT. SOFUT is used to get a listing of the SOF table of contents to verify that the mass matrix was read onto the SOF.

9,10. The utility OUTPUT1 reads K1 & M2 onto the NASTRAN file INPT.

APPENDIX E

DMAP MODIFICATIONS TO DIRECT TRANSIENTS

This appendix supplements the description of DIRECT TRANSIENT ANALYSIS Solution Strategy. The Rigid Format 9 DMAP ALTER statements are given first, followed by explanations according to statement number.

```
 1.  ALTER     2,2
 2.  FILE      KGGX = TAPE/KGG = TAPE/UDVT = APPEND/
               TOL=APPEND $
 3.  ALTER     30,30
 4.  ALTER     33,33
 5.  PARAM         //C,N,ADD/V,N,NOBGG=-1/C,N-1/C,N,O $
 6.  ALTER     34
 7.  COND      LBL1, NOSIMP $
 8.  ALTER 67
 9.  INPUTT1       /,,,,/C,N,-1/C,N,O $
10.  INPUTT1       /K1,M1,,,/C,N,O/C,N,O $
11.  EQUIV         K1, KGG/NOBGG/M1, MGG/NOBGG $
12.  ALTER         102
13.  PURGE         MAA/NOKGGX $
14.  ALTER         110
15.  JUMP          LBL5 $
16.  ALTER         113
17.  PURGE         K4AA/NOKGGX $
18.  ADD           KAA,/K4AA/C,N,(0.03, 0.0)$
19.  CHKPNT        K4AA $
20.  ALTER         139
```

```
21.  PARAM       //C,N,ADD/V,N,NOSIMP/C,N,1/C,N,0 $
22.  ALTER       163,163
23.  EQUIV       PPT, PDT/NOSET $
24.  EQUIV       PDT, PD/PDEPDO $
25.  ALTER       167
26.  PARTN       UDVT, USTRIP,/,,FLITOU,/C,N,7/C,N,1/C,N,2/C,N,2/
                 C,N,2/C,N,2 $
27.  OUTPUT1     ,,,,//C,N,-1/C,N,4 $
28.  OUTPUT1     FLITOU,,,,//C,N,0/C,N,4 $
29.  SDR2        CASEXX, CSTM, MPT, DIT, EQDYN, SILD,,, BGDT, TOL,,,
                 EST,, PPT/OPP1,,,, C,N,TRANRESP $
30.  SDR3        OPP1,,,,,/OPP2,,,,, $
31.  CHKPNT      OPP2 $
32.  OFP         OPP2,,,,,//V,N, CARDNO $
33.  SAVE        CARDNO $
34.  EXIT        $
35.  ENDALTER
```

Statement Explanations:

1,2. The TOL (TRANSIENT OUTPUT LIST) must be enabled to that it can be appended for any continuation of integration. Initially, when there is no restart, the TOL need not be appended, but for all subsequent restarts it needs to be appended, so it is enabled here to be ready for all runs. Instead of inserting a separate APPEND statement, it becomes more concise to add the APPEND to the existing FILE statement. Consequently, the existing file statement was removed and put back with the added APPEND statement.

4.5 Subsequently it is desired to incorporate a structural damping matrix and no other kind of damping, but to equip the code via a DMAP ALTER and not through a matrix generator module. Ordinarily the EMG module would sense the absence of viscous damping and automatically set the parameter NOBGG to negative one so that the viscous damping matrix, BGG, would not be generated. But in this solution path EMG will be bypassed by the conditional jump of statement 7 (because stiffness and mass matrices are input via statements 9, 10, 11). Something must be done about parameter NOBGG so the OSCAR can do a proper job in providing storage space for input data blocks. NOBGG must be preset to -1, so the parameter entry C,N,1 must be changed to C,N,-1. The original PARAM statement must be replaced by this revised statement therefore the ALTER 33,33 was used.

3 thru 7. All matrix data is being read in from disc and no matrix generator will be called upon. The compiler recognizes this and sets NOSIMP to -1 during execution of TA1. This is all very well but if the sequence of operations were left unchanged NOSIMP would cause the conditional jump of statement 30 to engage statement #62 next and would completely bypass the PARAM statement #33 in spite of all the modifications discussed above. Consequently, the conditional jump based on NOSIMP was taken out (position 30 with the ALTER 30,30, and restored after the PARAM executes in position 33 and needs to have a value of +1 later on.

8 thru 11. Stiffness and mass matrices from Phase II need to be introduced into the DMAP stream after the matrix operators and before the matrix partitioners implying after statement #6.5,

SMA3, and before statement #75, GSPS. Here, ALTER 67 brings it in just before GP4.

The matrices on file FLITKM from Phase II can be internally subscripted to any variety of numbers from K1, M1 to K9, M9 and higher depending on the execution order of commands in Phase II. In anticipation of this, statements 10 and 11 use subscript one followed by equivalences to KGG and MGG which will always "take" because parameter NOBGG was preset to -1. If K & M are written out with other than subscript one, the acting values should be written into statements 10 and 11.

12 and 13. If there were OMIT's the SFA (Segment File Allocator) could not provide for MAA, because it was purged back in statement #28. This purge at #28 took place in this instance, because NOSIMP is negative as explained above. In order to equip SFA to provide for SMP2, MAA has to be unpurged before statement #103 when SMP2 goes into operation; therefore the unpurge is introduced at #102 with the control parameter NOKGGX which was set equal to +1 at statement #31.

14 and 15. It is the intention to provide for the damping matrix K4AA by a DMAP ALTER subsequently, besides which K4FF does not exist to act as an input data block for SMP2; therefore SMP2 and CHKPNT are jumped around.

16, 17, 18 and 19. Now the damping will be generated. The output data block name will be K4AA. But this was purged along with MAA at statement #28 so it also has to be unpurged; again the choice of control parameter is NOKGGX Spatially

uniform structural damping is proportional to stiffness so a simple scalar multiply will serve. The ADD statement performs $\alpha A + \beta B = c$. If B=0, then C=αA. The coefficient α on A is a complex number, demanding that the real and imaginary parts be supplied. In this instance, the real part is 0.03 and the imaginary part is 0.0.

20 and 21. KDEK2 is a parameter which indicates to module GKAD whether or not matrices KAA and MAA are present according to its value of +1 or -1. Its value is computed in statement #140 by a logical AND operation on the values of NEGENL and NOSIMP from their most recent update according to the VPS table. Unless something is changed at this point NOGENL and NOSIMP are both = -1 so their AND result = -1 and KEDK2 will be negative. This negative value would indicate that KAA and MAA do not exist, but they do exist, because matrices were read in without benefit of general elements or from element matrix generators. A non-negative value reflects the condition of these matrices. By changing NOSIMP to +1, the computed value of KDEK2 is +1, satisfying the non-negative requirement.

23 and 24. EQUIV statement #163 has a bug in it. To correct this bug, the two EQUIV statements are written in place of the defective statement.

26, 27, and 28. These statements provide for stripping velocity and acceleration vectors from the UDVT matrix. The partitioning vector USTRIP is supplied by the analyst via DMI card data. The displacement vectors in UDVT occur in columns 1, 4, 7, ... , the last output displacement. The last output

displacement is calculated to be:

$$\frac{160\left(\begin{array}{l}\text{the number of integration}\\ \text{time steps selected on}\\ \text{the T STEP card}\end{array}\right)\times 3\left(\begin{array}{l}\text{the number of vec-}\\ \text{tors per time step}\\ U,\ \dot{U},\ \ddot{U}\end{array}\right)}{8\left(\begin{array}{l}\text{the output interval}\\ \text{selected on the T STEP card}\end{array}\right)}+3\left(\begin{array}{l}\text{3 vectors}\\ \text{for the}\\ \text{zeroeth}\\ \text{time step}\end{array}\right)$$

$= \frac{160\times 3}{8} + 3 = 63$. This value is put in field 8 of the DMI card indicating that the DMI vector for controlling the columns partitioning will have 63 rows. The matrix of displacement vectors will be named FLIT #U, where # indicates the restart number starting with the zeroeth.

29 thru 35. The call to SDR2 allows the modules to respond to the Case Control Card OLOAD. Modules are limited to the input and output data blocks necessary to process only the loads. Exit after OFP because no plots are requested.