THE AUTOMATED MULTI-STAGE SUBSTRUCTURING SYSTEM FOR NASTRAN

E. I. Field, D. N. Herting,
D. L. Herendeen and R. L. Hoesly

Universal Analytics, Inc.
Playa del Rey, California

SUMMARY

The new substructuring capability developed for eventual installation in Level 16 is now operational in a test version of NASTRAN. Its features are summarized. These include the user-oriented, Case Control type control language, the automated multi-stage matrix processing, the independent direct access data storage facilities and the static and normal modes solution capabilities. A complete problem analysis sequence is presented with card-by-card description of the user input.

INTRODUCTION

35

One of the most desired improvements in NASTRAN has been the capability for automated, multiple-stage substructuring analysis. The substructuring method reduces the difficulty of analyzing large and complex structures by dividing the analysis into small, more manageable tasks. Substructuring is a logical extension of the basic finite element method itself. That is, each component substructure is but a complex, finite part of the whole. This concept is easily extended to include the idea of combining substructures which are themselves combinations of component substructures. This process is called multi-stage substructuring.

The complex computational tasks of identifying the characteristics of each component, joining these components to form the final full model, and managing the associated data involve sophisticated computer program requirements beyond the existing scope of NASTRAN. Although the NASTRAN program currently provides the basic tools needed to perform simple substructuring analyses, the use of these capabilities requires significant experience and extensive intervention on the part of the user. Therefore, a new approach was proposed by Universal Analytics, Inc. This approach was presented by the NASTRAN Systems Management Office (NSMO) to a team of potential major aerospace users for their review and qualification to assure the viability and utility of the concepts proposed. Based on their review, the following final design criteria were established:

1. Analysis of large problems with a facility for unlimited multi-stage combinations of substructures.

2. Performance of both static and normal mode analyses (Rigid Formats 1, 2, and 3) at any stage of substructure combination with flexibility for extension at a later date to other Rigid Formats.

3. Execution on all three of the main frame computers on which NASTRAN is currently maintained (IBM, UNIVAC, CDC).

4. Elimination of all arbitrary restrictions on the sequencing of grid points and on the use of coordinate systems in defining the basic substructures.

5. Repeated application of the same basic substructure data for identical or symmetrical subcomponents of a model without redefinition of that model.

6. Communication of substructuring data between any two of the three main frame computers.

7. Simple user control by the novice while retaining existing NASTRAN flexibilities for the expert.

Each of these criteria was met with a minimum of machine-dependent programming using the following basic design features:

1. A bulk storage direct access file, independent of the standard NASTRAN file structure, was established for the Substructure Operating File (SØF) for storing all substructuring matrix and control data between each phase of processing.

2. A Master Data Index (MDI) file, also stored on the SØF, was designed to provide identification and control over all data sets on the SØF. A simply connected tree structure was selected to define all substructure component relationships and to provide unique trace-back facilities for retrieval of solution data at the basic substructure level.

3. A substructure Control Deck system of commands, using linguistic constructs, patterned after the current Case Control Deck, was developed for simple control over all steps of the analysis. To provide this feature, each command is automatically translated during execution into a set of DMAP instructions which are inserted as alters to the requested Rigid Format.

4. New Bulk Data Card options were provided so that all references to data contain only original basic substructure names and grid point identifiers.

The following sections provide an overview of how the system was implemented. The key user features are discussed and tables are included which list the substructuring commands and the Bulk Data options. A complete analysis sequence for a simple problem is illustrated in the appendix with a full card-by-card description of the input.

DESIGN SPECIFICATIONS

The specifications that were developed according to the criteria outlined above for implementing multi-stage substructuring into NASTRAN were based on a fully automated processing procedure. The alternative of enhancing the DMAP approach already available in NASTRAN was discarded to avoid: 1) the inherent requirement for user involvement in detail file maintenance, vector definition and matrix manipulations, 2) the overly constricting limitations on modeling of the substructures, and 3) the necessity of being an expert in NASTRAN to use this approach. Though the test system for the fully automated approach was implemented first in a Level 15.5 of NASTRAN, it was designed to minimize the effort of incorporation into Level 16.

The basic theory for the substructuring method is well known. The details of that theory to define each of the processing steps in NASTRAN need not be reviewed here. Substructure processing includes building the matrices for each of the basic substructures, defining the coordinate and matrix transformations needed for connecting two or more rotated, translated, or reflected component substructures, performing matrix reduction and solution, and computing the inverse transformations to recover solution results at any stage of the analysis. NASTRAN already provides a full set of modeling tools to generate the basic substructure matrices. It also provides most of the elementary matrix processing and output generating modules via the DMAP instruction set. The principal tasks were to develop the substructuring modules required to provide:

1. User control capabilities.

2. Data management features to maintain a Substructure Operating File (SØF) for storage and retrieval of substructuring data.

3. Program control for the execution of matrix operations requested by the user.

The user facilities provided by the new system are summarized below, followed by an overview of the SØF file maintenance utilities and of the methods used to control NASTRAN for automated substructuring.

USER CONTROL CAPABILITIES

A substructure analysis is performed in three phases. In Phase 1, the "basic substructures" are generated using the existing NASTRAN modeling data card input for elements, grid points, and constraints, etc. In Phase 2, various basic substructures may be "combined" and/or "reduced" in several steps to produce a "solution structure". Phase 2 also includes solution processing as well as the recovery of solution data for any previously defined level of substructure combination. In Phase 3, the basic Phase 1 processing is restarted using the checkpointed data or by resubmitting the original input data to obtain detailed displacement, force, and stress output for that basic substructure.

Each phase is run as a separate job step. Though the user may request all Phase 2 steps be performed in one execution, he will usually elect to subdivide the Phase 2 processing into several runs with each execution spanning one or more steps. This allows for examination and checkout of the intermediate results. This approach offers the following advantages:

1. Each component model of the overall structure (e.g., wing, fuselage, engine nacelles, landing gear, etc.) may be developed independently, even by separate contractors and on separate computer hardware systems.

2. Larger component substructures may themselves be assembled from yet smaller component substructures for multi-stage substructure analyses.

3. Each component substructure may be validated independently, plotted and analyzed prior to assembly and solution of the integrated whole model.

4. Changes due to errors, model modifications, and/or design alterations may be effected for any basic substructure and reintegrated into the overall structure at a minimum cost.

5. Via matrix reduction of the stiffness and mass matrices of neighboring substructures, their interaction effects on any given component can be economically included in the separate analysis of that particular component.

The user exercises control over the substructure operations via the "Substructure Control Deck" which contains a set of commands for directing the basic operations in each phase of the analysis. A summary of these commands and their associated subcommands is given in Table 1. The detailed data for defining transformations, connectivities, boundaries, constraints, etc., are input by the user via the new Bulk Data cards, summarized in Table 2. As can be seen the Substructure Control Deck options provided full control over each step in the analysis, selective output at each step, ample visibility into the contents of the SØF file, and simple management facilities to control the storage, purging, and retrieval of SØF data files. A detailed card-by-card description of the input for a simple problem is presented in the appendix to illustrate the convenience and simplicity of the system.

Each command uses terminology related to the operation performed. The primary operations of REDUCE, CØMBINE, SØLVE, and RECØVER can be requested in any order desired by the user. The user is relieved of the tedious and error-prone tasks involved in keeping track of the matrices, partitioning vectors, internal numbering sequences, and details of the coordinate geometries, etc. Connections of component substructures may be found automatically or they may be specified manually. The component substructures which occur repeatedly may all be equivalenced to one component substructure and rotated, translated, or reflected into their respective positions in the final model. Undeformed plots may be requested at any step. If severe errors are detected, input checking on the remaining steps is performed and the time consuming matrix operations are skipped.

TABLE 1. SUMMARY OF SUBSTRUCTURE COMMANDS

A. Phase and Mode Control	
SUBSTRUCTURE #	- Defines execution phase (1, 2, or 3) (Required)
ØPTIØNS	- Defines matrix options (K, M, or P)
RUN	- Limits mode of execution (DRY, GØ, DRYGØ, STEP)
ENDSUBS #	- Terminates Substructure Control Deck (Required)
B. Substructure Operations	
CØMBINE	- Combines sets of substructures
NAME	- Names the resulting substructure
TØLERANCE*	- Limits distance between automatically connected grids
CØNNECT	- Defines sets for manually connected grids and releases
ØUTPUT	- Specifies optional output results
CØMPØNENT	- Identifies component substructure for special processing
TRANSFØRM	- Defines transformations for named component substructures
SYMTRANSFØRM	- Specifies symmetry transformation
SEARCH	- Limits search for automatic connects
EQUIV	- Creates a new equivalent substructure
PREFIX*	- Prefix to rename equivalenced lower level substructures
REDUCE	- Reduces substructure matrices
NAME*	- Names the resulting substructure
BØUNDARY*	- Defines set of retained degrees of freedom
ØUTPUT	- Specifies optional output requests
SØLVE	- Initiates substructure solution (statics or normal modes)
RECØVER	- Recovers Phase 2 solution data
SAVE	- Stores solution data on SØF
PRINT	- Stores solution and prints data requested
BRECØVER	- Basic substructure data recovery, Phase 3
SPLØT	- Initiates substructure undeformed plots
C. SØF Controls	
SØF #	- Assigns physical files for storage of the SØF (Required)
PASSWØRD	- Protects and insures access to correct file
SØFØUT or SØFIN	- Copies SØF data to or from an external file
PØSITIØN	- Specifies initial position of input file
NAMES	- Specifies substructure name used for input
ITEMS	- Specifies data items to be copied in
SØFPRINT	- Prints selected items from the SØF
DUMP	- Dumps entire SØF to a backup file
RESTØRE	- Restores entire SØF from a previous DUMP operation
CHECK	- Checks contents of external file created by SØFØUT
DELETE	- Edits out selected groups of items from the SØF
EDIT	- Edits out selected groups of items from the SØF
DESTRØY	- Destroys all data for a named substructure and all the substructures of which it is a component

Manditory Control Cards

* Required Subcommand

TABLE 2. SUBSTRUCTURE BULK DATA CARD SUMMARY

A. Bulk Data Used for Processing Substructure Command REDUCE

BDYC	- Combination of substructure boundary sets of retained degrees of freedom
BDYS	- Boundary set definition
BDYS1	- Alternate boundary set definition

B. Bulk Data Used for Processing Substructure Command CØMBINE

CØNCT	- Specifies grid points and degrees of freedom for manually specified connectivities - will be overridden by RELES data
CØNCT1	- Alternate specification of connectivities
RELES	- Specifies grid point degrees of freedom to be disconnected - overrides CØNCT and automatic connectivities
GTRAN	- Redefines the output coordinate system grid point displacement sets
TRANS	- Specifies coordinate systems for substructure and grid point transformations

C. Bulk Data Used for Processing Substructure Command SØLVE

LØADC	- Defines loading conditions for static analysis
MPCS	- Specifies multipoint constraints
SPCS	- Specifies single point constraints
SPCS1	- Alternate specification of single point constraints
SPCSD	- Specifies enforced displacements for single point constraints

At each step in the analysis, the user identifies by name, e.g., HUB, WING, ØØT, etc., each substructure to be used in that step. All specific references o grid points for connection or boundary sets, releases, and loads, etc. are ade with respect to the basic substructure name. The names of any component ubstructure can be used for the combine, reduce, equivalence, solve, and re- over operations. Automatically the program retrieves all the relevant data or the named substructures from the SØF, performs the matrix operations re- uested, and stores the results on the SØF. Thus, the user is freed from the edious task of bookkeeping. If the same component substructure is to be used ore than once, e.g., identical components are to be used to create the full model, he "equivalence" operation must be used to assure unique names are assigned to ach substructure and its contributing components.

Several features have been provided for input data checking. Principal mong these is the DRY run option. This option allows the user to submit his un to have the program validate the consistency of his command structure and is data without actually performing the more time consuming matrix operations. lso available is a STEP option which first checks the data and then executes he matrix operations one step at a time. If errors are detected in the data, he matrix operations are skipped and the remainder of the processing sequence is xecuted as a DRY run only.

A second feature provided allows the user to process only selected matrix lata. For example, if the user finds that after having assembled his solution structure he wishes to add new loading conditions, or he wishes to obtain normal modes but did not have the mass matrix, he may re-execute the sequence of matrix operations to process only the load or only the mass matrix.

A third feature is available for displaying all the relevant substructuring data generated by the program. The data items that can be printed automatically are listed in Table 3. Using the output options provided, the user can verify explicitly each and every connectivity. If desired, the user may also obtain lists of all the retained degrees of freedom of the resulting pseudostructure to verify the completeness and accuracy of his input. These are all identified by basic substructure grid point numbers.

The processing for any one analysis can be carried out across all three computer systems (CDC, IBM, and UNIVAC). That is, the SØF data file created on one computer may be written to magnetic tape and shipped to another center for processing on any of the three standard hardware systems. This facility allows for several contractors to participate in a cooperative analysis of complex structures using their own computer centers.

DATA MANAGEMENT

The key to data management for the new automated substructuring system is the Substructure Operating File, the SØF. This one file is structured to hold all the relevant information for each component substructure. All the data items required for any basic or component substructure are listed in Table 3. Also

TABLE 3. SØF DATA ITEMS REQUIRED FOR EACH COMPONENT SUBSTRUCTURE

Item Name	Description
EQSS	External grid point (using basic substructure IDs) and internal point equivalencing data including scalar indices and associated components for each internal point number
BGSS	Defines the geometric coordinates and local coordinate system ID for each internal point of a component substructure in terms of the basic coordinate system for that component
CSTM	Contains the coordinate transformation data for every local coordinate system referenced in the BGSS item
LØDS	Directory of set IDs for all loads on each contributing basic substructure defined in Phase 1
PLTS	Names of each contributing basic substructure and its basic coordinate system transformation data to be used in generating undeformed plots
SØLN	Contains either static solution vector identifiers by subcase or eigenvalue and eigenvector parameters
KMTX	Stiffness matrix
MMTX	Mass matrix
PVEC	Load vectors
PØVE	Load vectors on points omitted during matrix reduction
UPRT	Partitioning vector used in matrix reduction
HØRG	H or G transformation matrix
UVEC	Displacement vectors or eigenvectors
QVEC	Reaction force vectors

stored on the SØF are the Master Data Index (MDI) file which serves as the directory for each substructure, the Director Index Table (DIT) which contains the names of each component substructure in the MDI, and the NXT array which serves to chain together all the data blocks available on the SØF. To have access to any item of data on the SØF, only the item substructure names are required.

The SØF is a permanent file physically stored on a user disk pack, drum, or equivalent device. It is constructed as a direct access file to avoid long and costly searching. It is used to communicate the data between all different phases of a multi-level substructuring problem and is maintained independently from the usual NASTRAN files. This choice was made to avoid what would have been a severe overload of the existing NASTRAN facilities. For example, since each substructure requires at least 6 data blocks, a practical limit for the old NASTRAN facilities would have been reached with as few as 30 substructures. Table 1 lists the commands provided the user with which he can maintain and protect his data on the SØF.

A full set of utilities is provided to maintain the SØF as well as to store and retrieve specific data items as they are required by the processing modules. Though the SØF is considered to be a single logical file, it may be physically stored on one to ten devices. This feature provides the user with an open-ended file capability which may be extended dynamically as the analysis progresses and more space is required. It therefore serves as a combination data block pool and a checkpoint file between job steps.

By interrogating the MDI, the DIT and the NXT, the SØF utilities provided can be used to:

1. Create or destroy a substructure
2. Delete items associated with a substructure to recover from errors
3. Equivalence substructures
4. Randomly locate in the file selected items associated with any substructure
5. Read and write items on the SØF
6. Dump and restore data to tapes as backup, or permanent storage, or as overflow to reduce the number of physical files required at any one execution.

A significant additional capability has also been provided, unique in the history of NASTRAN. The SØF data created on one computer may be written to tape, shipped to another center, and read into a different computer. This inter-computer communication capability allows for construction of complex structural system models from substructures developed by different contractors, at widely separated locations and even on different computers.

PROGRAM CONTROL

The user exercises primary control over the execution of his analysis with the Substructure Control Deck commands listed in Table 1. One of the eleven new modules developed especially for substructuring, ASDMAP, processes the Substructur Control Deck. Its design allows for future adaptation of the substructuring concepts to processing with other Rigid Formats than the three provided with the current system, Rigid Formats 1, 2, and 3.

The central concept of ASDMAP was to translate each primary command and its related subcommands into appropriate DMAP ALTERS to the Rigid Format being executed. The ASDMAP module was designed to recognize the various matrix and dry run options, tabulate and check the substructure names, and diagnose the substructure control deck for user errors.

In the actual implementation, each of the substructure commands is interpreted with the aid of built-in block data tables. For each substructure operation, the basic DMAP statements, the allowable subcommands, the optional cards, and the entries to be changed in the DMAP sequence all are stored in the form of simple control tables. These tables are then used to direct the program execution The ASDMAP module reads the cards associated with a command. It then generates the ALTER and DMAP card images, merges these with user-specified ALTER cards, and writes the merged set on the existing XALTER file, a logical file on the problem tape. The experienced user of NASTRAN retains the full flexibility of modifying and adapting the DMAP sequence produced to meet his own specialized requirements.

Inherent in the philosophy behind the design for the ten remaining substructu modules was the concept of an independent system using the NASTRAN subroutines for convenience while minimizing interaction with the remainder of NASTRAN. For this purpose, all of the substructure modules and the SØF file utilities were isolated to link 9 of NASTRAN. The number of input and output NASTRAN data blocks in the modules has been kept to a minimum, using the SØF files for the majority of data storage. In addition to the normal rules and restrictions on NASTRAN module design, the basic criteria for substructure modules were:

1. The NASTRAN matrix utility subroutines should be used for all matrix operations. This maintains compatibility with possible changes in future levels of the system.

2. Machine-dependent coding should be kept to a minimum. Only the initial file allocation for the SØF and the one SØF input/output routine are machine dependent.

3. The SØF file is basically a storage device and is not directly accessible by the NASTRAN utility routines. Matrix data from the SØF should be transferred to a scratch file before using the NASTRAN matrix utilities

4. The format of each SØF data item should be kept independent of the level of combination; i.e., a basic substructure is assumed to be a combination substructure made up of only one component.

5. In addition to the NASTRAN parameters, certain control options may be stored on the internal Case Control data block. In particular, this method is used for passing the extensive control data required by the CØMBINE and REDUCE operations in Phase 2.

6. If a non-trival error in the data is detected, the DRY run parameter should be set and all possible effort should be made to complete the execution of that module. When the DRY run parameter is "ON", all further time consuming matrix operations should be skipped and every attempt should be made to check user input data and control parameters.

Extensive documentation has been added to the NASTRAN Programmer's Manual for every subroutine, data block, and file structure that was developed and/or modified for automated substructuring.

CONCLUDING REMARKS

The automated substructuring capability described above has been installed and tested on all three major hardware systems, CDC, UNIVAC and IBM. It will be implemented in NASTRAN Level 16.

The test system is being verified at a number of facilities across the country. Several fixes have been made since its delivery and certain enhancements have been suggested by its users. These enhancements are being considered in preparing the specifications for Level 16 installation.

With the support of these users who have been willing to experiment with the new system, who have delved into the code and offered specific coding suggestions, and who have reported their performance timing history, the automated multi-stage substructuring system will become a welcome and reliable addition to NASTRAN.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Dr. D. J. Weidman of NSMO, for sponsoring this development project, and to Mr. H. P. Adam and Dr. E. C. Stanton of Douglas Aeronautics Company West, for monitoring its program under contract 6-73-182H with Universal Analytics, Inc.

APPENDIX

The following example illustrates an entire simple substructuring analysis. Figure A1 shows the basic substructures, TABLE and LEGS. Each has a loading specified; each has a different basic coordinate system; and each uses the same grid point identifiers. Figure A2 shows how these two components can be combined to make the final model using the reflective symmetry option.

The complete data decks to generate and analyze this structure are listed in Tables A1-A4. These include the data for generating the basic substructures in Phase 1, the assembly of the complete structure, solution, and data recovery in Phase 2, and the data recovery in Phase 3. A card-by-card explanation of their input demonstrates the simplicity of the new NASTRAN automated multi-stage substructuring system.

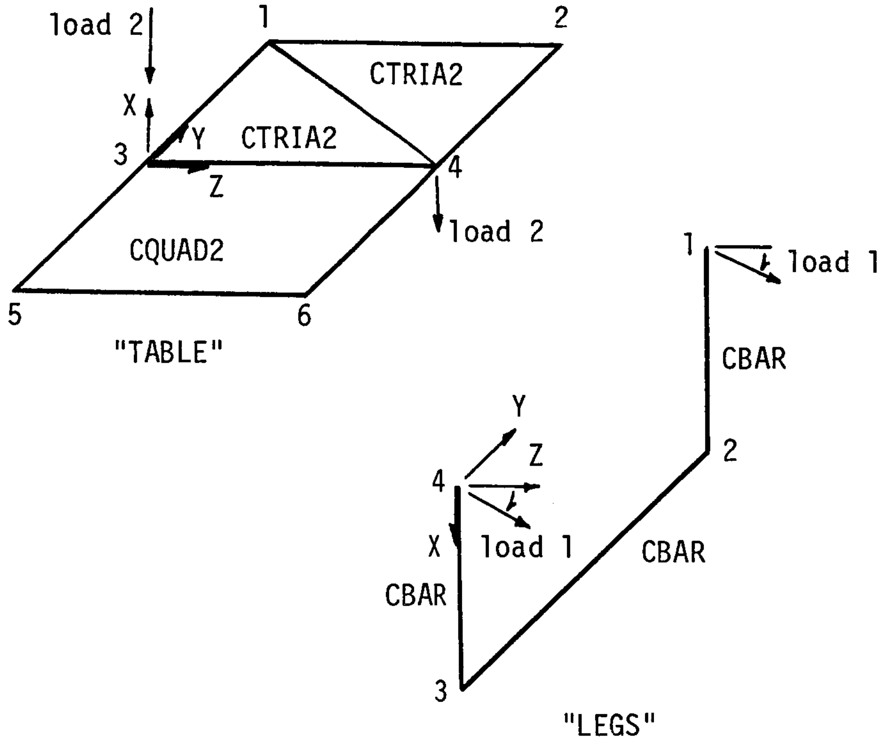

FIGURE A1. PHASE 1 - BASIC SUBSTRUCTURES (TABLE AND LEGS)

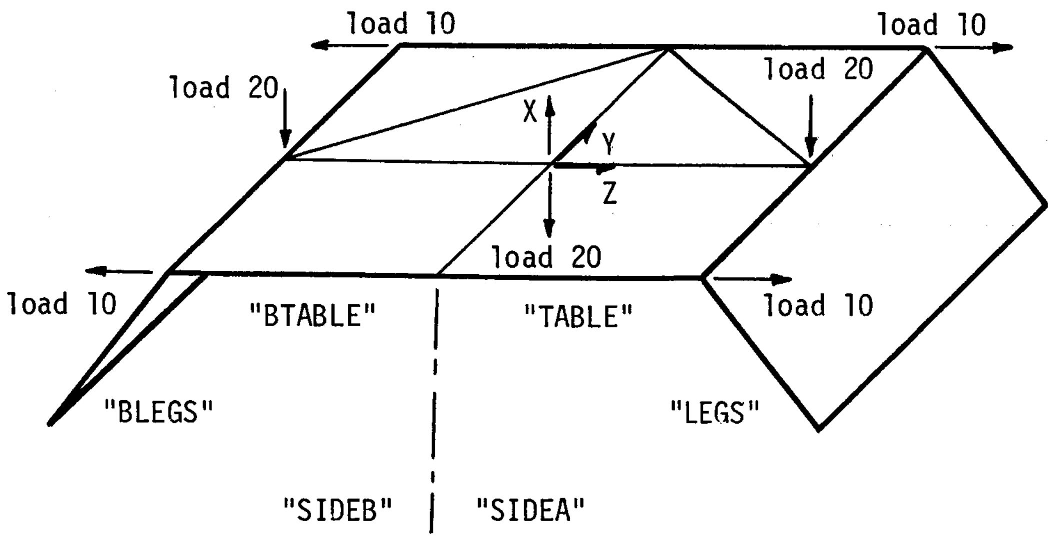

FIGURE A2. PHASE 2 - COMBINED SUBSTRUCTURE

TABLE A1. PHASE 1 DATA DECK FOR SUBSTRUCTURE TABLE

Card No.

```
 1  ID TABLE,BASIC
 2  APP DISP,SUBS
 3  SØL 2,0
 4  TIME 1
 5  CHKPNT YES
 6  CEND

 7  SUBSTRUCTURE PHASE1
 8  PASSWØRD=PRØJECTX
 9  SØF(1)=SØF1,250,NEW
10  NAME=TABLE
11  SAVEPLØT=1
12  SØFPRINT TØC
13  ENDSUBS

14  TITLE=TABLE, PHASE ØNE
15  LØAD=2
16  ØUTPUT(PLØT)
17  SET 1=ALL
18  PLØT
19  BEGIN BULK
```

Card No.	1	2	3	4	5	6	7	8	9	10
20	CQUAD2	3	2	5	6	4	3			
21	CTRIA2	1	1	1	2	4				
22	CTRIA2	2	1	3	4	1				
23	FØRCE	2	3		10.0	-1.0				
24	FØRCE	2	4		10.0	-1.0				
25	GRID	1		0.0	0.0	5.				
26	GRID	2		0.0	7.	5.				
27	GRID	3		0.0	0.0	0.0				
28	GRID	4		0.0	7.	0.0				
29	GRID	5		0.0	0.0	-5.				
30	GRID	6		0.0	7.	-5.				
31	GRID	7						123456		
32	MAT1	1	3.+7		.3	4.3				
33	PQUAD2	2	1	.1						
34	PTRIA2	1	1	.1						
35	ENDDATA									

TABLE A2. PHASE 1 DATA DECK FOR SUBSTRUCTURE LEGS

```
Card
 No.

  1  ID LEGS,BASIC
  2  APP DISP,SUBS
  3  SØL 2,0
  4  TIME 1
  5  CHKPNT YES
  6  CEND

  7  SUBSTRUCTURE PHASE1
  8  PASSWØRD=PRØJECTY
  9  SØF(1)=SØF4,7500
 10  NAME=LEGS
 11  SAVEPLØT=1
 12  SØFØUT INP3
 13     PØSITIØN=REWIND
 14     NAME=LEGS
 15  EDIT(32) LEGS
 16  ENDSUBS

 17  TITLE=LEGS PHASE ØNE
 18  LØAD=1
 19  ØUTPUT(PLØT)
 20     SET 1=ALL
 21     PLØT
 22  BEGIN BULK
```

Card No.	1	2	3	4	5	6	7	8	9	10
23	CBAR	1	1	1	2	5			2	
24	CBAR	2	1	3	2	5			2	
25	CBAR	3	1	4	3	5			2	
26	FØRCE	1	1		2.0	3.0	.0	4.0		
27	FØRCE	1	4		2.0	3.0	.0	4.0		
28	GRID	1		0.0	10.	0.0				
29	GRID	2		5.	10.	0.0				
30	GRID	3		5.	0.0	0.0				
31	GRID	4		0.0	0.0	0.0				
32	GRID	5		100.	100.	0.0		123456		
33	MAT1	1	3.+7		.3	4.3				
34	PBAR	1	1	1.0	50.	100.	10.			
35	ENDDATA									

TABLE A3. PHASE 2 DATA DECK (CØMBINE, REDUCE, SØLVE, AND RECØVER)

```
Card
 No.

  1  ID SUBSTR,PHASE2
  2  APP DISP,SUBS
  3  SØL 1,0
  4  TIME 1
  5  DIAG 23
  6  CEND

  7  SUBSTRUCTURE PHASE2
  8  PASSWØRD=PRØJECTX
  9  SØF(1)=SØF1,250
 10  ØPTIØNS=K,M,P
 11  SØFIN INP3,TAPE
 12     PØSITION=REWIND
 13     NAME=LEGS
 14  SØFPRINT TØC
 15  CØMBINE LEGS,TABLE
 16     NAME=SIDEA
 17     TØLER=0.001
 18     ØUTPUT=1,2,7,11,12,13,14,15,16,17
 19     CØMPØNENT LEGS
 20        TRANS=10
 21  EQUIV SIDEA,SIDEB
 22     PREFIX=B
 23  CØMBINE SIDEA,SIDEB
 24     NAME=BIGTABLE
 25     TØLER=0.001
 26     ØUTPUT=1,2,7,11,12,13,14,15,16,17
 27     CØMPØNENT SIDEB
 28        SYMT=Y
 29  REDUCE BIGTABLE
 30     NAME=SMALTABL
 31     BØUNDARY=100
 32     ØUTPUT=1,2,3,4,5,6,7,8
 33  SØFPRINT TØC
 34  PLØT SMALTABL
 35  SØLVE SMALTABL
 36  RECØVER SMALTABL
 37     PRINT BIGTABLE
 38     SAVE BTABLE
 39  SØFPRINT TØC
 40  ENDSUBS

 41  TITLE=PHASE TWØ SUBSTRUCTURE
 42  DISP=ALL
 43  SPCF=ALL
 44  ØLØAD=ALL
 45  SPC=10
 46  SUBCASE 1
 47     LØAD=10
 48  SUBCASE 2
 49     LØAD=20
 50  ØUTPUT(PLØT)
 51     SET 1=ALL
 52     PLØT
 53  BEGIN BULK
```

TABLE A3. (continued)

Card No.	1	2	3	4	5	6	7	8	9	10
54	BDYC	100	LEGS	20	BLEGS	20				+A
55	+A		TABLE	10	BTABLE	10				
56	BDYS1	1C	4	1	3	4	5			
57	BDYS1	1C	123456	2	6					
58	BDYS1	20	123456	2	3					
59	LOADC	10	1.0	LEGS	1	1.0	BLEGS	1	1.0	
60	LOADC	20	1.0	TABLE	2	1.0	BTABLE	2	1.0	
61	SPCS1	10	BLEGS	123456	2	3				
62	SPCS1	10	BTABLE	4	1	3	4	5		
63	SPCS1	10	LEGS	123456	2	3				
64	SPCS1	10	TABLE	4	1	3	4	5		
65	TRANS	10		.0	7.0	-5.0	3.0	11.0	-5.0	+B
66	+B	0.0	8.0	-5.0						
67	ENDDATA									

TABLE A4. PHASE 3 DATA DECK

```
Card
 No.

  1  ID TABLE,BASIC
  2  APP DISP,SUBS
  3  SØL 1,0
  4  TIME 1
  5  RESTART TABLE,BASIC (Restart deck)
  6  CEND

  7  SUBSTRUCTURE PHASE3
  8  PASSWØRD=PRØJECTX
  9  SØF(1)=SØF1,250
 10  BRECØVER BTABLE
 11  ENDSUBS

 12  TITLE=PHASE THREE FØR REFLECTED TABLE
 13  DISP=ALL
 14  ØLØAD=ALL
 15  SPCF=ALL
 16  STRESS=ALL
 17  SUBCASE 1
 18  SUBCASE 2
 19     LØAD=2
 20  BEGIN BULK
 21  ENDDATA
```

Phase 1 Data Deck for Substructure TABLE

Card No.	Refer to Table A1 for input cards described below.
1-6	Standard NASTRAN Executive Control Deck except the 'SUBS' option is selected on the APP card.
7	First card of Substructure Control Deck. Phase 1 is selected.
8	Password protection on the SØF is 'PRØJECTX'.
9	The SØF consists of one physical file with an index of one. (Indices must begin with one and increase sequentially.) The name of the file is 'SØF1' and it has a maximum size of 250,000 words. The file is to be initialized. (Internal pointers will be set to indicate that the SØF contains no data.)
10	The basic substructure to be generated will be identified by the name TABLE.
11	Plot set 1 will be saved on the SØF for performing plots of the combined structure in Phase 2.
12	Print a table of contents for the SØF. This includes a list of all substructures and their data items.
13	End of Substructure Control Deck
15	Selects the load to be saved on the SØF for use in Phase 2. Note that multiple loads may be saved by using multiple subcases. In addition to external static loads, thermal loads and element deformation loads may be selected.
16-17	Plot control cards are required if the SAVEPLØT subcommand is used in the Substructure Control Deck. These cards are used to define the plot sets for Phase 2 plotting. It is not necessary that a plot tape be set up in Phase 1.
19-35	Standard NASTRAN Bulk Data Deck. These cards define the mathematical model of the basic substructure.

Phase 1 Data Deck for Substructure LEGS

Card No.	Refer to Table A2 for input cards described below.
1-6	Standard NASTRAN Executive Control Deck except the 'SUBS' option is selected on the APP card.
7	First card of the Substructure Control Deck. Phase 1 is selected.
8	Password protection on the SØF is 'PRØJECTY'.
9	The SØF consists of one physical file with an index of one. (Indices must begin with one and increase sequentially.) The name of the file is 'SØF4' and it has a maximum size of 7,500,000 words. The file has been used previously as an SØF.
10	The basic substructure to be generated will be identified by the name LEGS.
11	Plot set 1 will be saved on the SØF for performing plots of the combined structure in Phase 2.
12-14	After substructure LEGS has been generated and saved on the SØF, it is copied out to user tape INP3.

Card No.	
15	All data items for substructure LEGS are removed from the SØF. (The substructure name remains in the SØF directory, however.)
16	End of Substructure Control Deck
18	Selects the load to be saved on the SØF for use in Phase 2. Note that multiple loads may be saved by using multiple subcases. In addition to external static loads, thermal loads and element deformation loads may be selected.
19-21	Plot control cards are required if the SAVEPLØT subcommand is used in the Substructure Control Deck. These cards are used to define the plot sets for Phase 2 plottina. It is not necessary that a plot tape be set up in Phase 1.
22-35	Standard NASTRAN Bulk Data Deck. These cards define the mathematical model of the basic substructure.

Phase 2 Data Deck

Card No.	Refer to Table A3 for input cards described below.
1-6	Standard NASTRAN Executive Control Deck except the 'SUBS' option is selected on the APP card. DIAG 23 requests an echo of the automatic DMAP alters generated.
7	First card of the Substructure Control Deck. Phase 2 is selected.
8,9	These cards specify the same SØF used in Phase 1 for substructure TABLE.
10	The card causes matrix operations to be performed on stiffness, mass, and load matrices. The default for Rigid Format 1 is stiffness and loads only. However, Rigid Format 2 was selected in the Phase 1 decks. This caused all three matrix types to be generated in Phase 1.
11-13	Basic substructure LEGS is copied to the SØF from user tape INP3.
14	Print the SØF table of contents.
15-20	Perform an automatic combination of substructures TABLE and LEGS. The resultant combined pseudostructure will be named SIDEA. The tolerance for conenctions is 0.001 units. Detailed output is requested. The basic coordinate system for substructure LEGS is transformed according to transformation set 10 in the Bulk Data.
21,22	Create a new secondary substructure SIDEB which is equivalent to SIDEA. This operation causes image substructures BLEGS and BTABLE to be generated.
23-28	Perform an automatic combination of substructures SIDEA and SIDEB. The resultant combined pseudostructure will be named BIGTABLE. The tolerance for connections is 0.001 units. Detailed output is requested. The basic coordinate system for pseudostructure SIDEB is symmetrically transformed about the XZ plane, identified by Y, the axis normal to the plane (sign change for all 'Y' degrees of freedom).
29-32	Perform a matrix reduction on the matrices of substructure BIGTABLE. The resultant reduced pseudostructure will be named SMALTABL. The retained degrees of freedom are selected in boundary set 100 in the Bulk Data. Detailed output is requested.
33	Print the SØF table of contents.
34	Plot pseudostructure SMALTABL. The plot control cards in the Case Control Deck are referenced.

Card No.	
35	Perform a static solution of pseudostructure SMALTABL. The constraint sets and load sets selected in the Case Control Deck are used.
36-38	Recover the displacements of substructures BIGTABLE and BTABLE from the solution of SMALTABL and save then on the SØF. Also, print the results for substructure BIGTABLE. The output requests in the Case Control Deck are referenced when the PRINT subcommand is invoked.
39	Print the SØF table of contents.
40	End of the Substructure Control Deck
42-44	Case Control output requests. Referenced by the PRINT subcommand of the RECØVER command.
45-49	Constraint and load set selections are referenced by the SØLVE command.
50-52	Plot control cards are referenced by the PLØT command.
54-58	These Bulk Data cards define the boundary set of retained degrees of freedom which was selected in the REDUCE operation (cards 29-32).
59-64	These cards define the loads and constraints selected in the Case Control Deck for the substructure SØLVE operation.
65,66	These cards define the transformation which is applied to the basic coordinate system of substructure LEGS in the first CØMBINE operation (cards 15-20).

Phase 3 Data Deck for Substructure BTABLE

Card No.	Refer to Table A4 for input cards described below.
1-6	Standard NASTRAN Executive Control Deck except the 'SUBS' option is selected on the APP card. "Card" 5 is actually the Restart deck punched out in Phase 1 for substructure TABLE
7	First card of the Substructure Control Deck. Phase 3 is selected.
8,9	These cards specify the same SØF used in Phase 2.
10	This card causes the data for the image basic substructure BTABLE to be copied from the SØF to GINØ data blocks. The data can then be used for data recovery operations, i.e., deformed structure plots, stresses, etc.
11	End of Substructure Control Deck.
13-16	Output requests for Phase 3 data recovery.
17-19	The subcase definitions in Phase 3 must be identical to those used in the SØLVE operation in Phase 2. SPC and MPC constraints in Phase 3 must be the same as those used in Phase 1. Load sets selected in Phase 3 must correspond to those selected in Phase 2 for each subcase. However, load sets selected in Phase 2 which do not exist for this particular basic substructure can not be selected in Phase 3.