NASA Conference Publication 2505

Sixteenth NASTRAN® Users' Colloquium

Proceedings of a colloquium held in Arlington, Virginia April 25-29, 1988

NASA

NASA Conference Publication 2505

Sixteenth NASTRAN® Users' Colloquium

Proceedings of a colloquium held in
Arlington, Virginia
April 25–29, 1988

National Aeronautics
and Space Administration

**Scientific and Technical
Information Division**

1988

FOREWORD

NASTRAN® (NASA STRUCTURAL ANALYSIS) is a large, comprehensive, nonproprietary, general purpose finite element computer code for structural analysis which was developed under NASA sponsorship and became available to the public in late 1970. It can be obtained through COSMIC® (Computer Software Management and Information Center), Athens, Georgia, and is widely used by NASA, other government agencies, and industry.

NASA currently provides continuing maintenance of NASTRAN through COSMIC. Because of the widespread interest in NASTRAN, and finite element methods in general, the Sixteenth NASTRAN Users' Colloquium was organized and held at the Quality Hotel, Arlington, Virginia on April 25-29, 1988. (Papers from previous colloquia held in 1971, 1972, 1973, 1975, 1976, 1977, 1978, 1979, 1980, 1982, 1983, 1984, 1985, 1986 and 1987 are published in NASA Technical Memorandums X-2378, X-2637, X-2893, X-3278, X-3428, and NASA Conference Publications 2018, 2062, 2131, 2151, 2249, 2284, 2328, 2373, 2419 and 2481.) The Sixteenth Colloquium provides some comprehensive general papers on the application of finite element methods in engineering, comparisons with other approaches, unique applications, pre- and post-processing or auxiliary programs, and new methods of analysis with NASTRAN.

Individuals actively engaged in the use of finite elements or NASTRAN were invited to prepare papers for presentation at the Colloquium. These papers are included in this volume. No editorial review was provided by NASA or COSMIC; however, detailed instructions were provided each author to achieve reasonably consistent paper format and content. The opinions and data presented are the sole responsibility of the authors and their respective organizations.

CONTENTS

		Page
FOREWORD		iii
1.	ON BULK DATA CARDS PROCESSING by Gordon C. Chan (Unisys Corporation)	1
2.	EXPERIENCES WITH A NASTRAN TRAINER by H. Grooms, P. Hinz and K. Cox (Rockwell International)	12
3.	DESIGN OF FEATS, A FINITE ELEMENT APPLICATIONS TRAINING SYSTEM by Alex Bykat (University of Tennesee at Chattanooga)	21
4.	ENHANCING YOUR COSMIC NASTRAN USAGE WITH PATRAN by Laurie C. Bender and Malcolm P. Johnson (PDA Engineering)	31
5.	EXPERIENCES WITH THE QUAD4 ELEMENT FOR SHELL VIBRATIONS by Melvyn S. Marcus, Gordon C. Everstine and Myles M. Hurwitz (David Taylor Research Center)	39
6.	COUPLED MASS FOR PRISMATICAL BARS by T. G. Butler (Butler Analyses)	44
7.	STRUCTURAL OPTIMIZATION WITH ROCKWELL NASTRAN by Viney K. Gupta (Rockwell International)	64
8.	EFFECT OF ELEMENT SIZE ON THE SOLUTION ACCURACIES OF FINITE-ELEMENT HEAT TRANSFER AND THERMAL STRESS ANALYSES OF SPACE SHUTTLE ORBITER by William L. Ko and Timothy Olona (Ames Research Center, Dryden Flight Research Facility)	79
9.	STRESS AND VIBRATION ANALYSIS OF RADIAL GAS TURBINE COMPONENTS by Ravi S. Krishnamurthy (Tiernay Turbines, Inc.)	128
10.	TREATMENT OF STATIC PRELOAD EFFECTS IN ACOUSTIC RADIATION AND SCATTERING by Gordon C. Everstine (David Taylor Research Center)	138
11.	A MAGNETOSTATIC NONLINEAR MODEL OF A ROTATING ARMATURE PRINTHEAD by T. J. Sheerer (Texas Instruments Incorporated)	153

CONTENTS
(Continued)

12. ARTIFICIAL INTELLIGENCE AND NASTRAN: VE-1 --- AN R.B.E.S. INTRODUCING NASTRAN 175
by V. Elchuri
(Aerostructures, Inc.)

ON BULK DATA CARDS PROCESSING

by

Gordon C. Chan
UNISYS Corporation
Huntsville, Alabama

SUMMARY

A significant speed improvement in processing NASTRAN bulk data cards, in the order of 10 to 40 times faster, has been achieved, as compared to COSMIC 1986 and 1987 NASTRAN releases. This improvement is directly proportional to the NASTRAN problem size. The improvement represents typically a 20% to 35% savings of time and cost on a normal NASTRAN job. In this project, a new XSORT2 module was written to replace the original XSORT module, which handles all the bulk data cards, and the old bulk data cards from the OPTP file. The XREAD routine that reads the input bulk data cards from the system input stream required major changes. The RCARD routine that interprets all characters in an input card and determines their type (BCD, numeric, or blanks) required minor changes for improved efficiency. Although the RCARD routine is not used in XSORT2, its changes increase the efficiency of the Input File Processor (IFP) module, which contributes to the overall efficiency of NASTRAN LINK 1, where the speed timing is checked.

XSORT2 is a completely new module with completely new logic, a new sorting technique, a new filing system, and a new data base management method. It bears no resemblance to the original XSORT module, and does not use any of the original supporting routines. However, it does the same job with the same result much faster and better. (The original XSORT did fail in several test cases. For example, multi-level of restarts with delete cards did not work properly). XSORT2 also uses new logic to handle a large number of continuation cards efficiently, while the original XSORT is known to handle this situation poorly and to be very time consuming. The XSORT2 source program, written in machine independent Fortran, is much easier to read and to understand. All bit and byte shiftings, word maskings, and character manipulations are kept to a minimum.

The new XSORT2 module has been thoroughly field tested. It is now a default module in the COSMIC 1988 NASTRAN release, replacing the less efficient XSORT module. However, the XSORT2 is actually installed in the COSMIC version in parallel with the original XSORT module. A user can invoke the original XSORT module by simply including a "DIAG 42" card in his NASTRAN input deck.

A NASTRAN job usually consists of an input phase of data processing, followed by a computational phase. The input data processing normally falls into three categories - the processing of the Executive Control cards, the Case Control cards, and the bulk data cards. A fourth category is required only if sub-structuring is requested. The Executive Control cards are processed by the Input File Processor, Part 1 (IFP1). The Case Control cards are handled by the Executive Case Control Processor (XCSA). The bulk data cards are processed by the XSORT module (Executive Bulk Data Card Sort), which reads and sorts the input bulk data deck. The input cards to the Executive and Case Control are completely free-field, and ordering independent. The bulk data cards are ordering independent and can be in fixed-field or free-field formats. This paper concerns only the bulk data cards, and the significant speed improvement.

All NASTRAN input cards are read into NASTRAN by the XREAD routine, which calls FFREAD to do the actual card reading from the system input stream. The following are the tasks that the XSORT module must handle when it processes a new bulk data card coming from XREAD:

Free-field vs. fixed-field format;
Single-field vs. double-field cards;
Cold-start vs. re-start;
Modified vs. unmodified restart;
Restart with or without delete;
Sorted and/or unsorted echoes, and punch;
Continuation cards and their parent cards;
Machine dependency, various word sizes, and word architectures;
Small vs. large input deck (where computer memory space is limited to handle all or part of the input cards in a single pass);
Data sorting and merge;
Mixed BCD and numeric data on input cards, and on output listing; and
User error checks, and error messages.

The input to NASTRAN can be considered quite user-friendly. User's errors will be flagged and messages printed out, but a NASTRAN job will not be terminated prematurely. However, it is not an easy matter internally to handle all the generous and flexible capabilities that NASTRAN allows during the input data processing phase. The complexity of the tasks involved can be gauged by the supporting subroutines that the XSORT module uses, which are listed below.

XREAD - calls FFREAD to read the input cards, in free-field or fixed-field formats;
XRECPS - positions the continuation cards to proper records;
RPAGE - a special page control routine;
INITCO - initializes machine dependent masks and constants;
XFADJ - calls XFADJ1 to adjust four character fields left or right, two or four fields at a time;
XBCDBI - converts BCD characters to binary integers for sorting;
XPRETY - "pretties-ups" BCD characters, integers, and floating point numbers for output printout;

CRDFLG - sets card type flags for restart;
EXTINT - converts card type field from machine dependent character code to an internal machine independent code - a process that is needed for alphanumeric sorting;
INTEXT - provides the reverse process of EXTINT;
ISFT - a special shifting function for XFAJ1;
KHRFNi - a group of four function routines for character-byte manipulation; and
GINO - a group of General Input and Output routines.

The complexity of the XSORT module is further complicated by the fact that the routines are highly machine sensitive, especially in the areas of bit and byte manipulations where different machines have different word sizes. The character function routines, KHRFNi (i=1,2,3,4), must be used for the VAX machine to bypass the word shifting and word masking difficulties (the VAX has different computer word architecture). Because of their simplicity and well defined functions, the KHRFNi group of routines has been gradually migrated into the source code for the other three machines (IBM, CDC, and UNIVAC). Before 1985, all the KHRFNi functions were written in machine dependent Fortran. In 1985 and thereafter, the KHRFNi functions in COSMIC NASTRAN were standardized and made machine independent, by the use of internal file I/O technique.

SOURCE OF DEFICIENCY

The XSORT module reads and processes the bulk data input cards either from the system input stream or from the Old Problem Tape (OPTP), and outputs the information in an orderly sequence to the New Problem Tape (NPTP), to be processed later by the IFP (Input File Processor) module. XSORT processes each character on an input card (80 characters per card) and determines its proper type - BCD, blank, or numeric. The characters are then split, moved, re-positioned, re-combined, or substituted to form meaningful data. Since the raw data are initially stored in BCD form, and 4 characters per word, the character manipulation functions, KHRFNi, or the equivalent left/right word shifting and word masking, are frequently employed to decode or encode the information. On average, 20 to 150 encode and decode operations are needed for an input card. The machine independent versions of KHRFN1 and KHRFN4 (since 1985) are highly I/O bound and time consuming, and the XSORT module is 4 to 8 times slower than the pre-1985 release. The VAX machines, heavily reliant on the KHRFNi routines, are greatly affected by the change made in 1985, and the XSORT module runs relatively slower.

NASTRAN requires all input bulk data cards to be sorted. The original XSORT module examines (via bit and byte manipulations) each input card by its first, second, third, and possibly up to the 9th, fields, and sets up its record position pointer, with respect to the other input cards previously processed. Each time a new input card is read in, a chain reaction of setting and resetting pointers follows (plus bit and byte manipulations). Finally, when either all the input cards are read in, or the computer available core space is full, the bulk data cards, saved in the core space, are transferred to either a scratch file or NPTP file, in sorted order given by the pointers. This method of sorting at each input card level provides a means to process a

large number of input cards with limited computer core space. However, it is definitely not the best way available.

The scheme by which XSORT handles large bulk data decks is quite interesting, but not necessarily the most efficient. The input cards are read in by XREAD, saved in the open core space, and transferred to a scratch file later as described in the previous paragraph. The original XSORT module allocates three scratch files for this purpose. When two of the three scratch files have received a fresh batch of data, XSORT merges these two scratch files and moves the results to the third file. The first two scratch files are now free for reuse. One of these files is used again to receive fresh data, and the other one is used again for the next level of file merging. This procedure continues until all input cards are read in and an ENDDATA card is encountered. The last merged file now contains bulk data cards in sorted order. Finally, the contents of this last merge file are transferred to the NPTP with continuation cards properly inserted. This method of merging three files, with three GINO buffer spaces, works successfully and reliably. However, this method is highly I/O bound and therefore slow. One big shortcoming of this method is that the intermediate files are getting bigger and bigger, a longer and longer time is required to create the merged files, and the beginning part of the data is copied too many times. This method works satisfactorily only for input decks that are "not too big".

The original XSORT module is also known to be very slow when a large number of continuation cards is present. The problem seems to be I/O bound again. Another shortcoming of the original XSORT module is that it does not fully utilize the capability of GINO (General Input and Output package) to buffer in and buffer out short blocks of data efficiently. In XSORT, each 20-word input card image is written to a scratch file as a record (a record can be from 800 to 1600 words). This original carelessness makes XSORT/GINO I/O bound again, and wastes disk storage space in all machines except VAX. (VAX has a different GINO package.)

NEW XSORT2 WITH NEW LOGIC

The first attempt to improve the original XSORT module was to plug the holes where time is slipping out. This requires great understanding of the source program, including some sections of the source code poorly written or poorly documented and hard to understand. Difficulties were also encountered in many machine dependent areas involving word maskings, left and right word shiftings, word size, and bit and byte operations. It was finally realized that it was easier to write a completely new module to replace the original XSORT. New logic, new techniques, and new methods can be applied freely to the new product without fears of crashing with some existing old source code.

The new XSORT2 module is machine independent. It completely avoids the character manipulation routines KHRFNi. In fact, it does not use the old XRECPS, RPAGE, INITCO, XFADJ/XRADJ1, XBCDBI, XPRETY, CRDFLG, EXTINT, INTEXT, and ISFT supporting routines. The unsorted bulk data card echo is now moved to FFREAD where the card is actually read in from the system input stream. XSORT2 takes full advantage of the data already left adjusted (good for sorted bulk data echo) coming from FFREAD if the bulk data is in free-field format.

If the bulk data is in fixed-field form, XREAD left adjusts all input fields if and only if it is called by XSORT2. Since in XREAD the input raw data are available in both BCD form and character format, XREAD converts part of the bulk data into their equivalent internal numeric codes, a prerequisite for alphanumeric sorting (done in XSORT2). The XREAD routine also informs XSORT2 of the current input card type - regular bulk data, comment, restart delete, continuation, or an ENDDATA card. (This task was previously done by XSORT). The new XSORT2 uses only two additional subroutines for support services:

SORT2K - (An existing in-core sorting routine) to sort table by two key words, and
BISLC2 - A new routine similar to BISLOC, to search an entry of two keys in a given table.

XSORT2 loads all input cards in their card image forms, and their corresponding equivalent internal numeric codes, into the open core, except comment, continuation, and delete cards. At the end (either the open core is full, or an ENDDATA card is read) XSORT2 calls SORT2K to sort the cards in core, then save them all in one GINO record to a scratch file. This process is repeated until all input cards are read and processed. XSORT2 allows up to 30 scratch files to be used to receive incoming data. (For practical reasons, only up to 17 files can be used.) The continuation cards and delete cards are saved separately in two different scratch files.

If more than 10 scratch files are used in the above process, a 2-to-1 file merge follows. If more than 17 files are employed, a 3-to-1 file merge is done before final file merging and the creation of the NPTP file. This pre-merging of files is intended to save buffer space during the final file merge. However, if the number of continuation cards is within manageable size, this pre-merging of files is not needed.

Before the final merging of all scratch files, the entire core space is allocated to hold as many continuation cards as possible. The final file merge involves merging of all scratch files simultaneously and the insertion of the continuation cards to their designated parents, to form the NPTP file. To be consistent with the rest of NASTRAN program requirements, each input card image to NPTP is written as a 20-word short record. Before this final merging, however, all GINO files are written in large blocks (as large as the working space in the open core can hold). Finally, a check is made for any unused continuation cards. User's warning messages are printed out if they exist.

Appendix A gives a step by step description of the method used in the new XSORT2 module. It gives more detail about the open core space usage, the OPTP file, the pre-merging and final merging of the scratch files, the setting of the restart flag, and the redundant unused continuation cards.

CONCLUSION

The original XSORT module is slow, inefficient, wasteful of disk space, and makes NASTRAN LINK 1 costly to run. The new XSORT2 is ultra efficient, and is 10 to 40 times faster (as compared to 86/87 COSMIC NASTRAN release).

The new XSORT2 module has been field tested in all four machines (IBM, CDC, VAX, and UNIVAC). All 119 NASTRAN standard demonstration problems ran successfully with the new module. Other tests designed to check out restart and substructuring also ran successfully. A few tests with intentional input errors stopped at the end of LINK 1, and proper error messages were echoed out correctly. A few tests with large input decks, 8,000 to 15,000 cards and 1,500 to 2,500 continuation cards, ran very fast. The speed improvements can be translated into some 2 to 5 times faster if they were to be compared to the pre-85 NASTRAN releases. The new XSORT2 module makes LINK 1 run noticeably faster when NASTRAN is run interactively.

The XSORT2 module is now installed in the COSMIC 1988 NASTRAN release, replacing the less efficient XSORT module. It is presently installed in parallel with the original XSORT module, and a user can invoke the old XSORT module by simply including a "DIAG 42" card in his NASTRAN input deck.

The new XSORT2 and the original XSORT modules are completely interchangeable - that is, XSORT2 can work with the bulk data deck coming from an OPTP tape, which is generated by XSORT, and vice versa.

APPENDIX A

(To be inserted in the NASTRAN Programmer's Manual following pages 4.4-1 through 4.4-11.)

4.4 EXECUTIVE PREFACE MODULE XSORT2 (EXECUTIVE BULK DATA CARD SORT)

4.4.1 ENTRY POINT: XSORT2

4.4.2 Purpose

The function of XSORT2 is to prepare a file on the New Problem Tape containing the sorted bulk data. The operation of XSORT2 is influenced by the type of run. If a cold start, the bulk data is read from the system input stream (the User's Master File is not supported), sorted, and written on the New Problem Tape. If an unmodified restart, the bulk data is copied from the Old Problem Tape onto the New Problem Tape. If a modified restart, the bulk data is read from the Old Problem Tape, and cards are deleted and/or added in accordance with cards in the system input stream. Additionally, flags are set within restart tables for each card type changed in any way. Again, the sorted bulk data is written onto the New Problem Tape. A print of the unsorted and/or sorted bulk data is made on request. If a request is not made in a restart run, sorted bulk data is automatically printed. XSORT2 processes all data cards between the BEGIN BULK and ENDDATA cards in the input stream. Both cards must be present to properly bracket the NASTRAN bulk data deck. If a DIAG 42 card is included in the Executive Control Deck, module XSORT2 will be replaced by XSORT, an original NASTRAN module.

4.4.3 Calling Sequence

CALL XSORT2. XSORT2, a preface module, is called only by the Preface driver, SEMINT.

4.4.4 Method

Step 1. The open core in /ZZXSRT/ is divided into 3 GINO buffers and a work area, and 3 scratch files are used. XSORT2 reads (via XREAD and FFREAD; the latter also prints the unsorted data if requested) from the system input stream a card at a time. If the input card is a comment, XSORT2 skips to read another card. If the card is a continuation card, it is saved in the scratch2 file. If the card is a restart delete, its delete range is saved in the scratch1 file. If the card is an ENDDATA card, no more cards are to be read from the system input stream. If the card is a regular bulk data card, the card is saved in the work area. Four additional words, the internal numeric code of the first 3 fields (plus the 4th or 5th field in some cards) supplied by XREAD, and an in-core record pointer, are also saved. This process is repeated until (a) an ENDDATA card is read, or (b) the work area is full. If this is a restart run, all input cards of the regular type are flagged for restart operation.

Step 2. If the work area is full, or an ENDDATA card is read, the data in the work area is sorted by the four internal numeric code words and the entire work area, except the in-core pointers, is written to the scratch3 file in the sorted order. Steps 1 and 2 are repeated if necessary until all input cards are read in, and an ENDDATA card is encountered. On the second pass of Step 2, data in the work area, minus the in-core pointers, are written to

scratch4. On the third pass, scratch5 is used, and so on. For practical reasons, up to 17 scratch files can be used, which gives a capacity of roughly 35,000 input cards if an open core space of 50k words is used. The capacity is directly proportional to the available open core space. (The XSORT2 source code actually allows up to 30 scratch files.)

Step 3. If this is an unmodified restart run with no delete card and no new bulk data card, the bulk data cards in OPTP are read and transferred to the NPTP file. The rest of the XSORT2 operation is skipped.

Step 4. If this is a modified restart with delete, the work area in core is loaded with the delete ranges from scratch1. Scratch1 is closed and reopened for reuse. The bulk data cards are moved from OPTP to scratch1 with cards deleted as specified by the delete ranges. The deleted cards, or parents of the deleted continuation cards are flagged for restart operation. If this is a modified restart, with or without delete, all continuation cards from OPTP are transferred to the continuation file, scratch2. These continuation cards from OPTP are marked so that in final file merging in Step 9, their parents will be flagged for restart operation.

Step 5. This pre-merge step is needed only when (a) more than 10 scratch files are used in Step 2, and (b) the open core space is not big enough to hold simultaneously all continuation cards, GINO buffers, and scratch working arrays. If Step 2 uses 10 to 17 scratch files, every other two files (2-to-1) are merged to form a new file. If more than 17 files are used in Step 2, a 3-to-1 file merge is used. The total number of scratch files that contain input data is now reduced to n. If this pre-merge step is skipped, n is the original number of scratch files used in step 2.

Step 6. n in Step 5 is increased by 1 if this is a restart run.

Step 7. The open core in /ZZXSRT/ is reaccessed. It is now divided into n GINO buffers, n 24-word arrays, a table area, and a data area. The table area must be big enough to hold the first 2 words of all the continuation cards plus a pointer for each card. The data area must hold at lease 300 continuation card images (minus the first 2 words each) to make XSORT2 efficient.

Step 8. The table area and the data area in Step 7 is loaded with the continuation card data previously saved in scratch2. The first 2 words plus a pointer are saved in the table, and the remaining card image is saved in the data area in a location corresponding to the pointer. When the data area is full, this entire data area is copied as one block of records to a new scratch file. Loading of the continuation cards into the table area and the data area is repeated if needed. (If the data area is big enough to hold all the continuation cards, no new scratch file is generated.) When scratch2 is exhausted, the in-core sorter, SORT2K, is called to sort the table

Step 9. All the scratch files that hold the bulk data cards, and if applicable, the scratch1 file that holds the OPTP data, are ready for final file merge. A record of each file is loaded into one of the n 24-word arrays

in an orderly sequence. By comparing the last 4 words of each of the n arrays (these are the 4 internal numeric code words), the first 20 words of the smallest array are written out to NPTP. The array is then replenished by the next record from the same scratch file. If the record just written out to NPTP specifies a continuation card, the continuation table is searched via the BISLC2 routine, and the continuation card is picked up from the continuation work area, or from the new continuation scratch file. If the continuation card originated from OPTP, the parent card in NPTP must be flagged for restart operation. The continuation card in the continuation table is now marked as "used". Step 9 is repeated until all scratch files are exhausted.

Step 10. This final step checks and prints any continuation cards that are left "not used". The continuation cards of a "not used" continuation card are marked off to avoid redundant messages.

4.4.5 Subroutines

4.4.5.1 Subroutine Name: XREAD

1. Entry Point: XREAD
2. Purpose: It reads an input card, and left-adjusts all fields. If XREAD is called by XSORT2 (the 5th word in labeled common /XECHOX/ is non-zero), it converts the first three input fields (plus the 4th or 5th field in some card types) to a set of 4 internal numeric codes, that can be used for sorting. These 4 coded words are saved in labeled common /XSORTX/. XREAD calls FFREAD to actually read an input card from the system input stream. The input card can be in fixed-field or free-field format.
3. Calling Sequence: CALL XREAD (*n,BUF) See subroutine XREAD for more details.

4.4.5.2 Subroutine Name: SORT2K

1. Entry Point: SORT2K, A secondary entry point in SORT, an in-core sorter
2. Purpose: It sorts a table by first 2 key words.
3. Calling Sequence: CALL SORT2K (0,0,N1,N2,TABLE,LEN) See subroutine SORT for more details.

4.4.5.3 Subroutine Name: BISLC2

1. Entry Point: BISLC2
2. Purpose: Binary search to position a double word in a table using the first entry. (Same function as BISLOC, which is a single word search)
3. Calling Sequence: CALL BISLC2 (*n1,ID,ARR,LEN,KN,JLOC)

Where: n1 - Nonstandard return if ID is not found in the first entry in ARR
ID - Integers to locate as first double word of entry - two integers in ID(1) and ID(2) - input
ARR - Table to search - input

LEN - Number of words in each entry of the array ARR - integer - input
KN - Number of entries in ARR - integer - input
JLOC - Integer pointer to location of first double word in the entry.

4.4.6 Design Requirements

1. Data cards operated upon by XSORT2 must conform to the NASTRAN format for bulk data cards (ten eight-character fields per card for fixed-field input, or all fields separated by comma or blanks for free-field input). See section 2 of the User's Manual for details.

2. Data cards must contain only valid BCD key punch codes or blanks. Nonstandard multi-punched code (e.g., some IBM EBCDIC) will cause unpredictable results.

3. For IBM machine only, data cards can be punched in EBCDIC or BCD.

4. XSORT2 requires sufficient open core to contain three GINO buffers and a work buffer for at least 200 data cards (each data card requires twenty-five core locations). However, for a large input deck (15,000 cards or more, and a large number of continuation cards) up to 11 GINO buffers may be needed.

5. The continuation cards must fit into the core work area during final file merge. Each continuation card requires three core locations.

6. XSORT2 logic is not biased toward input that is already sorted. An ultra fast in-core sorter is used for input card preparation. The intermediate (if needed) and the final file merges are ultra efficient.

7. During initial input card preparation and for practical reasons, XSORT2 is limited to 20 scratch files. 17 of these files are used to store input card images. The number of card images per file is n, where

n = ((available open core space) - 3*(GINO buffers)) / 25.

At this initial preparation stage, only three GINO buffers are used.

4.4.7 Diagnostic Messages

XSORT2 can produce two categories of diagnostic messages. The first are termed USER messages and deal with bulk data card errors. The second are termed SYSTEM messages, which are generally fatal in nature and indicate serious I/O malfunctions.

XSORT2 message numbers include 201 through 216. All messages are listed and explained in section 6 of the User's manual.

EXPERIENCES WITH A NASTRAN TRAINER

By

H. Grooms, P. Hinz, and K. Cox

INTRODUCTION

Engineers entering today's world have a fundamental theoretical understanding of the finite element method but have virtually no practical experience with it. The difference between understanding the theoretical foundations of the finite element method and analyzing a real structure using a computer program can be substantial. The NASTRAN Trainer was developed to address the latter issue.

Many researchers (ref. 1, 2, 3, 5) have addressed the development of user-friendly finite element analysis and design tools, but training engineers to use these tools is still an issue. Sadd and Rolph (ref. 6) concluded that training engineers in the use of the finite element method could be accomplished by any of three ways:

1. Using traditional university training

2. Utilizing the increasing number of specialized seminars and short courses offered in finite element analysis

3. Developing a tailored in-house training program

Sadd and Rolph took the third option and established a 28-hour course (4 hours per week for 7 weeks).

Grooms, Merriman, and Hinz (ref. 4) presented the concept of a NASTRAN Trainer as an automated method for familiarizing engineers with applying the finite element method to structural analysis problems. The NASTRAN Trainer is one of the functional elements in the system shown in figure 1. The documentation module of this system is completely functional, while the adviser (used for debugging models) is in the test stage. This paper will explain the following:

1. The organization of the NASTRAN Trainer

2. Contents of the Trainer

3. Steps that a user follows

4. Users' observations and suggestions

5. Plans for other applications

ORGANIZATION AND PURPOSE OF THE TRAINER

The Trainer was developed as a stand-alone tool that an engineer could use at his own convenience and pace. The system was designed so that the user would need very little knowledge of the job control language or the operating system before he could sit down at a terminal and solve an example problem.

The Trainer is organized into three main modules: (1) Overview, (2) User's Guide, and (3) Problem Set. Figure 2 shows some of the details of each module. The user accesses these modules by using the primary menu. More details of the "NASTRAN Environment" sections are given in figure 3.

The typical sequence of events for a user is shown in figure 4. The Trainer has been planned so that it is very easy for a first-time user to get started on an example problem.

CONTENTS OF THE TRAINER

Ten example problems are contained in the Trainer. These examples and their salient features are summarized in table I. The problems, which range from a statically determinate, two-dimensional truss to a ring-stiffened cylindrical tank, are shown in figures 5 through 14.

USER EXPERIENCES

Since 1986, approximately 65 engineers have used the NASTRAN Trainer. The majority of these users were new graduates who had taken one or more finite element courses in school but who had almost no actual experience with NASTRAN. These users typically went through the set of ten problems in two months while also performing their regular work. Approximately 20 engineers were surveyed by use of the questionaire shown in table II. The percentages shown in the table indicate the responses. By using the program, the average user reduced his training time from 135 hours to 60.

Many of the comments were directed to the NASTRAN documentation. The comments made about particular example problems are being used to modify and improve the Trainer. The users' consensus was that the Trainer is a useful and effective tool that should be expanded.

EXTENSIONS AND OTHER APPLICATIONS

The ten example problems that are currently in the Trainer were chosen to familiarize the novice user with

1. Bar and rod elements
2. Beam elements
3. Geometric symmetry
4. Loading symmetry and antisymmetry
5. Boundary conditions and stability constraints

6. Plate elements

7. Plane stress

8. Grid fineness

9. Temperature and loading

10. Three-dimensional considerations

Two additional areas that are candidates for modules are substructuring and normal modes analysis. These would be advanced modules that users would only undertake after they had completed the basic module.

The substructuring module would deal with

- Single versus multiple level
- Sequence of joining substructures
- Data handling

The normal modes analysis module would cover

- Reduction of stiffness matrices
- Reduction of mass matrices
- Accuracy considerations
- Dynamic response

CONCLUSIONS

The NASTRAN Trainer has demonstrated that it is an efficient and effective training tool as well as an aid to productivity improvement.

REFERENCES

1. Woodward, W.S.; and Morris, J.W.: Improving Productivity in Finite Element Analysis Through Interactive Processing, Finite Elements in Analysis and Design. Vol. 1, no. 1, 1985.

2. Wilson, E.L.; and Holt, M.: CAL-80-Computer Assisted Learning of Structural Engineering, Symposium on Advances and Trends in Structures and Dynamics, Washington, D.C., Oct. 1984.

3. Ginsburg, S.: Computer Literacy: Mainframe Monsters and Pacman, Symposium on Advances and Trends in Structures and Dynamics, Washington, D.C., Oct. 1984.

4. Grooms, H.R.; Merriman, W.J.; and Hinz, P.J.: An Expert/Training System for Structural Analysis, ASME Conference on Pressure Vessels and Piping, New Orleans, LA., June 1985.

5. Ginsburg, S.: Self-Adapting Menus for CAD Software, Computers and Structures, Vol. 23, no. 4, 1986.

6. Sadd, M.H.; and Rolph III, W.D.: On Training Programs for Design Engineers in the use of Finite Element Analysis, Computers and Structures, Vol. 26, no. 1/2, 1987.

TABLE I. SUMMARY OF EXAMPLE PROBLEMS

Example	Description	Significant Features	Classical Solution Compares
1	Statically determine plane truss subjected to point load	Rod elements, stability constraints	Reactions, stresses, deflections
2	Beam simply supported on one end and fixed at the other subjected to point load	Bar elements	Reactions, stresses, deflections
3	Tapered beam fixed at one subjected to point load	Tapered beam elements	Reactions, stresses, deflections
4	Plane frame subjected to point load	Half-model, symmetric and anti-symmetric loads	Reactions
5	Simply supported square plate subjected to out of plane point load at center	Plate bending elements, quarter model	Stresses, moments, deflections
6	Plate with hole in center subjected to in-plane load	Plane stress, quarter model, fine grid around hole	Stresses
7	Beam fixed at both ends subjected to through the depth temperature difference	Temperature input	Reactions, stresses
8	Simply supported beam subjected to temperature pattern	Half-model, temperature distribution decomposed into symmetric and anti-symmetric parts	Reactions, stresses, deflections
9	Cylindrical shell subjected to hydrostatic loading	3D, simulation of curved surface using flat elements	Reactions, stresses
10	Cylindrical shell with ring frames, closed at both ends subjected to internal pressure	3D, self-equilibrating loading	Stresses, deflections

TABLE II. QUESTIONNAIRE FOR USER FEEDBACK

	Critique of NASTRAN Trainer			
1	Was using this system a worthwhile expenditure of your time?			
	a. Yes	(89%)	Undecided	(11%)
	b. No	(0%)		
2	How much total time would you estimate that you spent using the Trainer? 60 hours			
3	How much total time would you have spent (estimate) to gain this knowledge if the Trainer had not been available? 135 hours			
4	The number of examples was			
	a. Too few	(17%)		
	b. Too many	(6%)		
	c. About right	(77%)		
5	The system was			
	a. Too simple	(17%)		
	b. Too complicated	(6%)		
	c. About right	(77%)		
6	Could the Trainer be improved by adding other topics?			
	a. Yes	(67%)	Maybe	(11%)
	b. No	(22%)		
7	Which section, if any, should be expanded upon? (Wide variety of responses.)			
8	How often (average) did you invoke the NASTRAN documentation manual section?			
	a. Never	(44%)		
	b. 0-2 times/example	(22%)		
	c. More than 2 times/example	(34%)		
9	Was the NASTRAN documentation section useful?			
	a. Yes	(38%)	Never used it	(29%)
	b. No	(33%)		
10	How often did you use (average) the printed Cosmic or MSC NASTRAN manuals?			
	a. Never	(6%)		
	b. 0-2 times/example	(17%)		
	c. More than 2 times/example	(77%)		
11	Please add any additional comments you desire. (Responses vary from "great" to "give us more advanced problems.")			

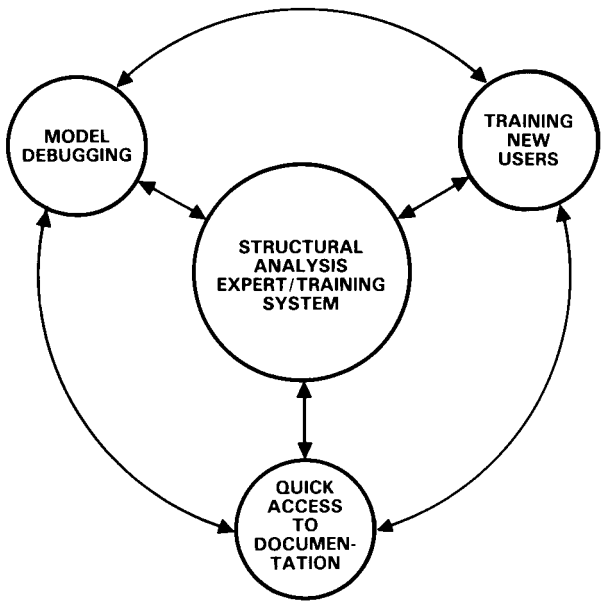


FIGURE 1. FUNCTIONAL EXPERT/TRAINING SYSTEM

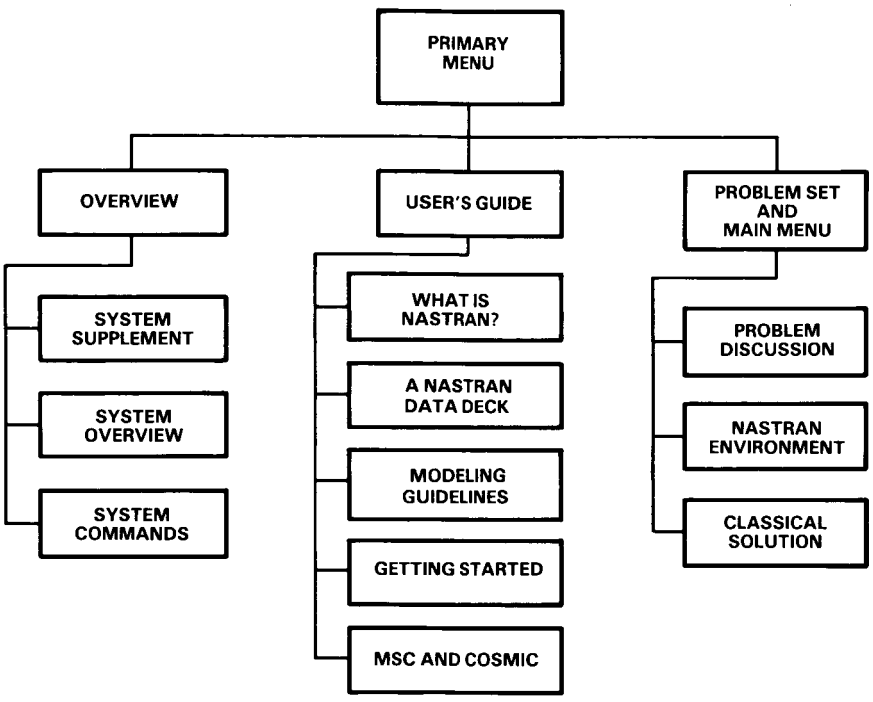


FIGURE 2. ORGANIZATION OF NASTRAN TRAINER

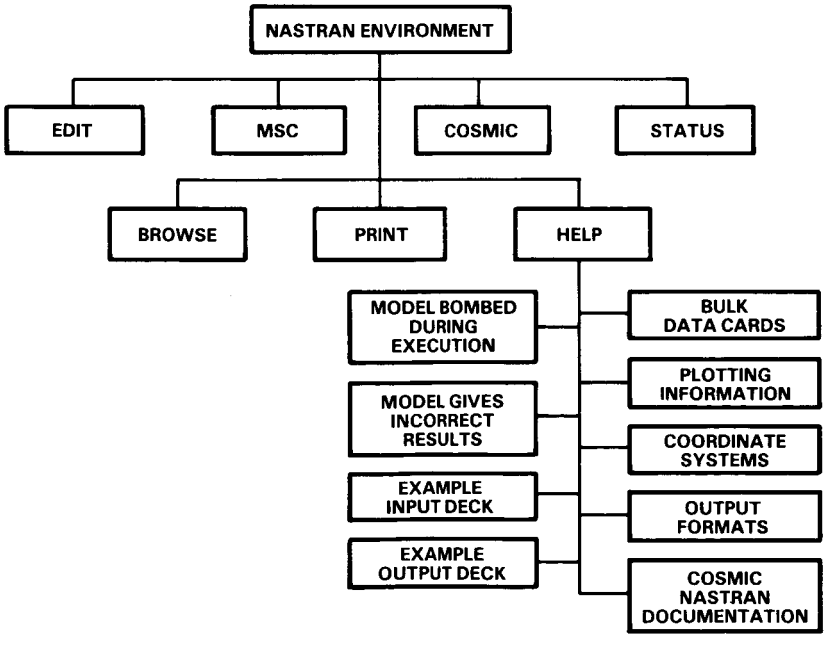


FIGURE 3. ORGANIZATION OF PROGRAM

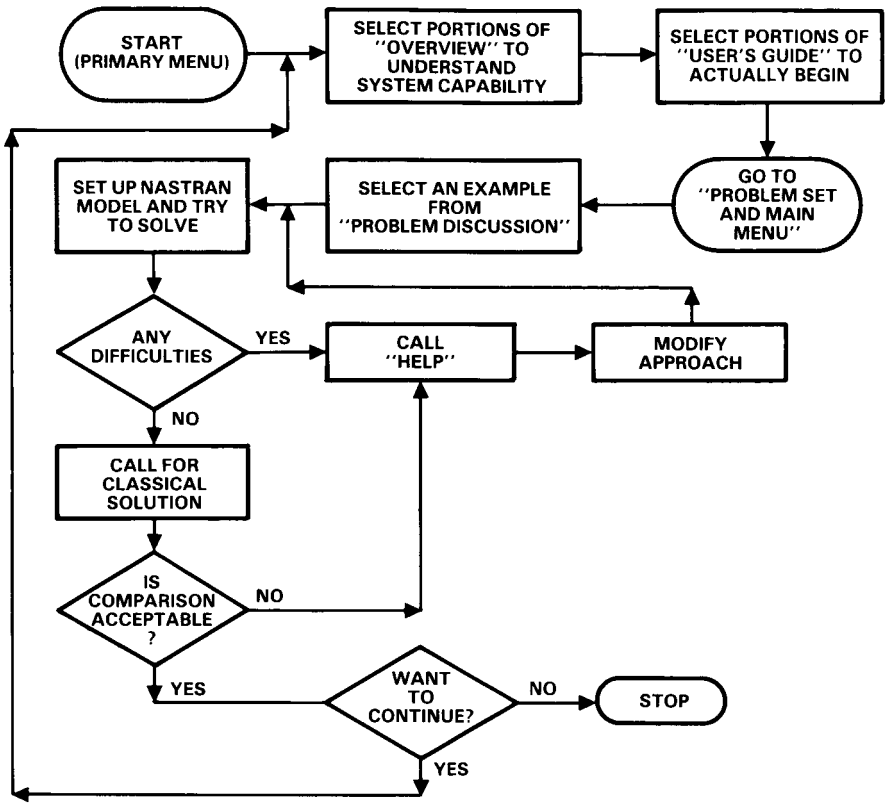


FIGURE 4. TYPICAL USER STEPS

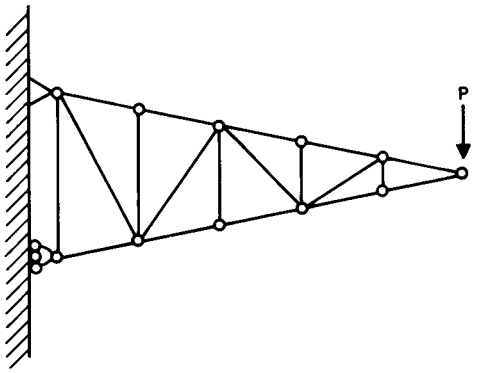


FIGURE 5. TWO DIMENSIONAL TRUSS (EXAMPLE 1)

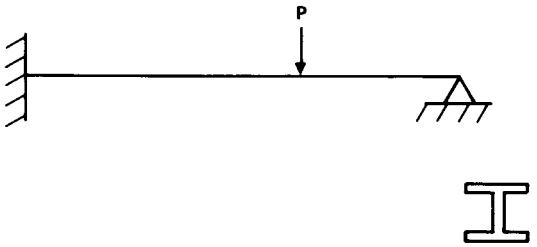


FIGURE 6. BEAM WITH POINT LOAD (EXAMPLE 2)

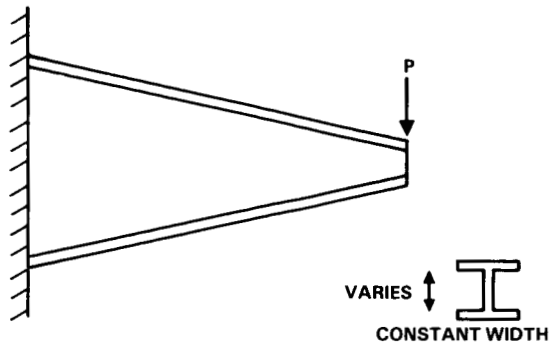


FIGURE 7. TAPERED BEAM SUBJECTED TO POINT LOAD (EXAMPLE 3)

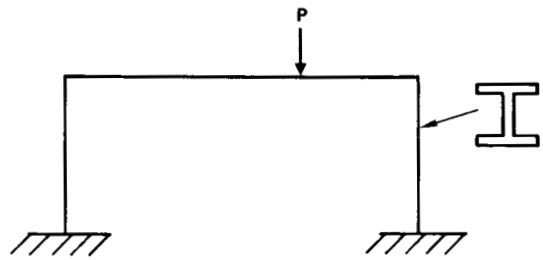


FIGURE 8. PLANE FRAME SUBJECTED TO POINT LOAD (EXAMPLE 4)

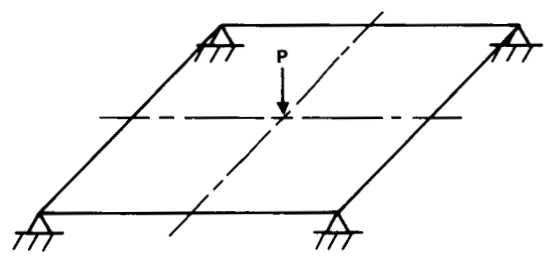


FIGURE 9. SIMPLY SUPPORTED SQUARE PLATE (EXAMPLE 5)

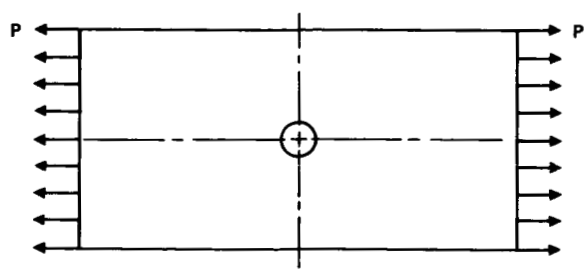


FIGURE 10. PLATE WITH HOLE IN CENTER (EXAMPLE 6)

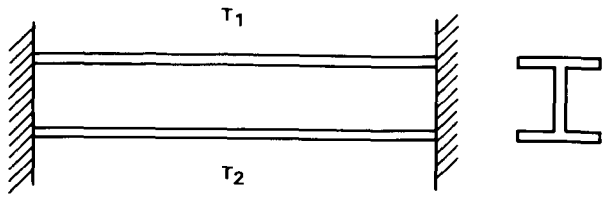


FIGURE 11. BEAM FIXED AT BOTH ENDS WITH TEMPERATURE LOADING (EXAMPLE 7)

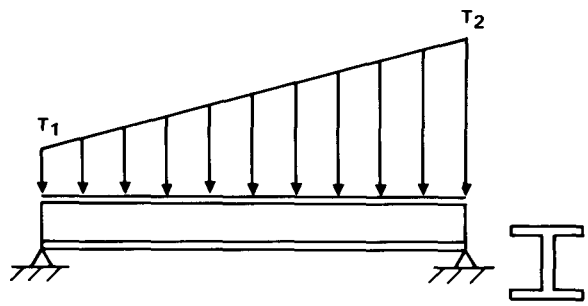


FIGURE 12. SIMPLY SUPPORTED BEAM SUBJECTED TO TEMPERATURE PATTERN (EXAMPLE 8)

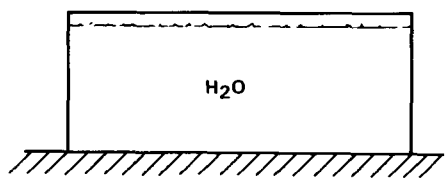


FIGURE 13. CYLINDRICAL SHELL SUBJECTED TO HYDROSTATIC LOADING (EXAMPLE 9)

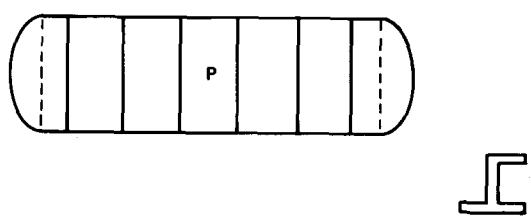


FIGURE 14. CYLINDRICAL SHELL WITH RING FRAMES SUBJECTED TO INTERNAL PRESSURE (EXAMPLE 10)

Design of FEATS, a Finite Element Applications Training System

Alex Bykat
Center of Excellence for Computer Applications
University of Tennessee at Chattanooga
Chattanooga, TN 37402

ABSTRACT

The Finite Element Method is a dominant numerical method which finds applications in fields such as aeronautics, structural engineering, reactor design, shipbuilding, geology, mining, to mention but a few. Due to the importance of its applications, a number of commercial packages have been written to implement the method and to make it available for engineering applications. Examples of such packages are Cosmic Nastran, MSC/Nastran, ANSYS etc. These packages are typically very large, very expensive, and require powerful and expensive computers.

Use of a finite-element analysis package requires highly trained engineers, possessing not only expertise in their professional area but also possessing knowledge of the inner structure of the software package. Further, this knowledge must be coupled with awareness of assumptions underlying the finite-element method implementation.

With continued tumbling of computer hardware costs, and concomitant reductions in software costs, it is the availability of such highly trained personnel that poses a barrier to widespread use of finite-element analysis.

This paper describes some aspects of our research project intended to make a breach in this barrier by constructing a knowledge based finite element applications consulting and training system (FEATS). The ultimate goal of FEATS is to test the proposed theories necessary to describe the functions of an intelligent system consultant and teacher in a finite element training environment. FEATS will be implemented on the TI EXPLORER LX. It will reside on the Explorer processor; the COSMIC NASTRAN finite-element package will reside on the LX side (an M68020 processor).

The pragmatic aims of FEATS are to create an interface to a Finite Element Package to offer intelligent features for control and interrogation of the underlying finite element system, as well as facilities for effective training of personnel in the use of the system resources. To perform its consulting/training functions FEATS will communicate in natural language and will use models of the user's knowledge, of the conversation, and of the domain. (The natural language interface is adopted from OSCAT [Bykat, 1986].)

THE PROBLEM.

Intimate knowledge of a sophisticated package requires a great deal of training, and many hours of practice coupled with the constant availability of a patient "guru". Unfortunately, whereas the novice user is (usually) willing to allocate the time

needed, and to put up with the training required, the guru is frequently not available and often is not at all patient.

A novice user, and indeed a more advanced one, may be easily discouraged by the constant need of experimentation which is frequently the only major alternative open as a complement to the very few hours at which the teacher is available. Indeed such an experimentation results also in a very inefficient utilization of human resources.

Manuals, be it on-line or not, are valuable but only as one of training options; they are of little merit when available as the only training tool. Using the on-line, or the hard copy, reference manual, the novice user is faced with masses of information to scan through. (For example, NASTRAN documentation has already over 8,500 pages!) Yet, frequently, the same information could be offered in 'no time' by an expert consultant. Furthermore, to avail himself even of this avalanche of facts, he must be sufficiently trained to be able to index his query with a correct keyword; incorrect keyword might at best retrieve no information at all, though more frequently it will simply swamp the user with irrelevant facts.

Customer service and 'hot lines' provided with expensive packages are helpful, but when used they do disrupt the application training process, turning frequently into discouragingly long procedures.

TOWARDS A SOLUTION - A WISH LIST.

As a consequence of such a situation, the need for an automated consulting system capable of training, answering, and explaining its answers to questions about the usage of the underlying system (and its domain) becomes apparent.

To be effective, the system should be unobtrusive, should support a mixed initiative dialogue, and should be able to measure the apprenticeship level of the student. Such a measure can then be used to choose a level of interaction which is appropriate to that particular student.

The system should also perform various other functions related to its consulting, training, and management of the underlying hardware roles. These functions require a model of the user (capturing his knowledge), a model of the machine (resources available), a model of the domain (FEP), and a model of the dialogue.

The capabilities of training functions to be investigated fall within the area of open problems in design of Intelligent Tutoring Systems. Much of the work in this field concentrates on construction of student models. Notable examples are GUIDON [Clancey, 1982], WUMPUS [Goldstein, 1982], SOPHIE I,II,III [Brown, 1982], and BUGGY [Burton, 1978]. Our work differs in the theories proposed. The main differences lie in the mechanism of knowledge collection and the calculus adopted for evaluation of the students knowledge and misconceptions.

A natural language interface is a requirement of great importance. Such interface, whenever appropriate, should use the graphics facilities offered by the system, to enhance the interaction with the user. This is of particular relevance in a finite element and training environment. In such applications, the expressive power of graphics input/output is a necessity.

TOWARDS A SOLUTION - THE FEATS PROJECT.

FEATS environment will consist of the components specified in Fig.1. FEATS will communicate with the Nastran finite element package (FEP) using the 'cooperating processes' paradigm, using direct streams as well as the shared memory protocols. When needed, the remote procedure call (RPC) protocol will be employed. All three methods are supported by TI EXPLORER LX hardware, which is used in development of FEATS.

FEATS unifies a number of cooperating modules including:

A. communication interface for input of user utterances and presentation of systems conclusions,
B. control module for rule construction, conflict resolution and rule invocation,
C. reasoning module for interpretation of possibly ill-formed user utterances, selection of appropriate rules, and explanation of conclusions reached,
D. model construction module for collection of facts and rules describing the user, his machine, and his conversation,
E. teacher module for instruction and training of concepts and facilities available under the underlying FEP system,

The knowledge base of the system will be programmed mostly into production rules. The concept of frames, [Minsky, 1975; Bobrow, 1977], is adapted to support the implementation of the models used by FEATS. Identification of these modules imposes a hierarchical structure which will be helpful in orderly implementation of this project. Figures 1, 2 and 3 show FEATS's architecture; various principles of the above modules are discussed below.

COMMUNICATION INTERFACE.

The OSCAT's NL interface prototype, [Bykat, 1986], is adopted for FEATS project. This interface performs as an expectation driven parser which processes each sentence as an individual unit. The sentences are parsed by using a dictionary of predefined words. Each word defines the expectation of other words which either precede it or follow it. The structure of the word definitions is fashioned after the Conceptual Dependency theory, [Schank & Abelson, 1975].

During parsing, the meaning of the sentences is formulated as a graph of linked word frames representing the semantic content of a sentence. Once the parse of the sentence has been terminated, the information acquired is then passed on to appropriate modules for further processing (identify goals, plan actions, generate response, etc).

Thus for example, a user utterance such as:

'I want to substructure this region into two parts.'

will be transformed by the NL interface into:

```
M1:  mood(talk).
G1:  goal(actor(U1), object(C1)).
A1:  mutate(actor(U1), object(E1), to(E2)).
N1:  config(rel(divided), object(E1), object(E2))
N2:  config(rel(part_of), object(P1), object(E2))
N3:  config(rel(part_of), object(P2), object(E2))
C1:  utter(act(A1), mod1(N1), mod2(N2), mod3(N3))
```

```
U1:  ‹user id›.
E1:  p_obj(id(S1), mod(_)).
E2:  p_obj(id(S2), mod(_)).
P1:  p_obj(id(S3), mod(_)).
P2:  p_obj(id(S4), mod(_)).
S1:  ‹description of a region›.          % exists
S2:  ‹description of a subdivision›.     % to be described
S3:  ‹description of a subregion›.       % to be described
S4:  ‹description of a subregion›.       % to be described
```

Notice the separation of the utterance into a number of concepts. Each of these concepts can be manipulated appropriately as the current focus of conversation warrants. Further, since these concepts are preserved, they can be referred to in subsequent conversation too.

Note also, that the main operation (action) is 'change an object' (ie. mutate). When 'mutate' is qualified by various nuances, it becomes 'substructure' (eg. N1), 'rotate', 'translate', 'shrink', etc.

REASONING MODULE.

The functions of the reasoning module are concerned with selection of rules which are appropriate for firing (invoking) in the current context. There are frequently a number of rules suitable for selection in any given situation. Conflicts can arise due to, the origin of two categories of rules, which are candidates for selection: (1) general rules inherited from the initial model of the FEATS world, and (2) specific rules selected by the pending goals as implied by the user's utterance. The reasoning module resolves all conflicts that arise.

Some of the more salient functions of this module are: goal extraction and plan formation. For example, the control module uses the internal representation of the conversation, to extract the goals and to create plans to satisfy these goals. Thus in the above example, the following goals will be extracted:

Formulate instructions to mutate a region.
Explain these instructions.

Note, how simple these important inferences are to obtain. This is achieved by a careful construction of the internal representation which in turn depends on dictionary definitions.

The training and the consulting aspects of FEATS require plan building. In this prototype we employ a hierarchical plan construction. Once the goal of the utterance is understood, the first level of the plan is established. The first level is then refined to produce a second level, the second level is refined to produce a level third, and so on.

Refinement of plans proceeds by invoking plan fragments which are pre-defined. On the other hand, composition of the plan fragments into subplans and whole plans depends entirely on the particular goal that is extracted from the utterance.

Thus, for example, for the goal "create Object", FEATS produces the following plan (indentation shows plan refinement):

```
start_plan
   create(Object)
     precond(create,Object)
        exists(Object,new)
        material(Object,Enough)
     use_tool(create,Object)
        identify_tool(create,Object,Tool)
        exists_tool(Tool,Id)
        apply_tool(Id,create,Object)
            use_method(Tool,create,Object,   Method)
            call(Method)
```

The interesting fact about the above plan is its generality. Thus, given the operation 'Operation' (eg. create), and the object 'Object', it requires only general search routines for the predicate exists and material to form a general model for performing the Operation on the Object.

The dependence on the domain of FEATS is thus isolated to specification of the Tool (looked up by the identify_tool predicate), discovery of the particular Tool's Id (in exists_tool predicate), and the specification of the method for using the tool (found by the use_method predicate). In the case of "create file" goal, these are specified in the knowledge base as:

```
file(create,editor).
     % to create a file use editor
editor('VI').
     % 'VI' is an editor
'VI'(create,file,[vi,FID]).
     % to create a file using VI
  % specify command:  vi <file id>
```

In addition to the above functions the reasoning module will perform, whenever requested, explanation of the conclusions reached by FEATS in satisfaction of user posed goals. This, of course, is of major importance for the training aspect of our project.

CONTROL MODULE.

A major function of the control module is the selection of rules applicable within the current context. Since the knowledge base is expected to grow into a considerable size, a crucial pragmatic concern for this module is its search efficiency.

To reduce the number of rules to be searched in any given instance, the knowledge base will be structured into classes of rules with each class declared as separate module. The search can then be restricted to a class of rules, subject to a particular set of goals, then within the class for a subclass of rules, subject to a particular subset of goals, etc. Other indexing structures will be considered.

MODEL CONSTRUCTION.

The following models will be created and used by the system:
(1) user model - includes: history of achievement, topics of deficiency, and a measure of apprenticeship level
(2) domain model - includes: machine resources, invariants and norms of the FEP system
(3) conversation model - includes: history of discourse, focus of current dialogue.

Construction and use of these models is intended to allow processing of possibly ill-formed user utterances, to select system's responses in the context of user apprenticeship level and conversation focus, as well as to guide the training process.

TEACHING MODULE.

FEATS will be designed to perform its evaluation actions unobtrusively. To achieve this we shall investigate an approach to gathering as much information for the user model as possible in a supervisory manner. That is, as the user interacts with the system, FEATS will gather information for the user model by carefully evaluating the user actions, much as a human supervisor would. This supervisory function will coexist with the test-and-grade (TAG) approach.

The supervisory function will extract (mainly negative) evaluation information from communication failures which attempt to violate the system model or the pragmatic beliefs of the system. The TAG function will yield (positive and negative) evaluation information by observing the effect of actions performed by the user under direction of FEATS.

Thus, two sources will supply data for the user model: the supervisory function, and the training TAG function. Information gathered in this model will then be used to select appropriate interaction level with the user.

CONCLUSION

This paper describes early stages of the FEATS project. FEATS offers intelligent features for control and interrogation of the underlying finite element system, as well as facilities for effective training of personnel in the use of the system resources.

A prototype of FEATS is being written in Prolog on a Texas Instruments Explorer LX. The latter is a dual processor machine consisting of a lisp machine (EXPLORER) and an M68020 based computing engine (LX) running a Unix System V. This provides therefore an ideal environment for cooperation between AI type of a system and an engineering type of a system. In our case, the AI system is FEATS, whereas the engineering system is NASTRAN.

APPENDIX 1: FINITE ELEMENT PRINCIPLES.

The Finite Element Method is a dominant numerical method used in the solution of

Partial Differential Equations over regions with irregular geometries. This method finds applications in many fields, eg: aeronautics, structural engineering, reactor design, shipbuilding, geology, mining, to mention but a few. A number of commercial finite element packages exist, eg. NASTRAN, NISA, ANSYS, etc. As a rule, these packages are large and complex. For example, the MSC/NASTRAN has over 480,000 lines of FORTRAN code, and over 15 volumes of documentation.

Essentially, the method consist of three main phases.

In the first phase the region of integration is subdivided into a number of (simplicial) elements, and over each such element a trial function is proposed. A trial function approximates the solution of the system over that element. The 'total' solution is then expressed as a sum of solutions over the elements of the region.

In the second phase, the total solution is formulated in terms of the trial functions (with prescribed continuity conditions). This phase, referred to as the 'assembly phase' results in a system of equations whose unknowns represent the values of the required solution at the nodes of the elements. Typically, the resulting equations are very large and sparse. The distribution of nonzeros in the equations is then condensed via node reordering, or element reordering.

In the third phase, the resulting equations are solved. In fact, the solution can be realized without the assembly phase. Such methods have a number of advantages, as well as disadvantages.

When the above three phase cycle is completed, the accuracy of the solution may require refinement of the subdivision (local or global), and the above solution process to be repeated over a new subdivision. To afford an automatic implementation of this refine-and-solve loop, the data structures representing the subdivision must be appropriately designed.

Some of the research by the author concerning the above stages of FEM is described in the following papers [Bykat, 1973; 1974; 1976; 1977; 1983].

REFERENCES.

Anderson J.R."Acquisition of proof skills in geometry" Michalski, Carbonell & Mitchell, 1983

Brown J.S., Burton R.R., de Kleer J."Pedagogical, natural language and knowledge engineering techniques in SOPHIE" in Sleeman and Brown, 1982

Brown, J.S, Burton, R.R, "Diagnostic models for procedural bugs in basic mathematical skills", Cognitive Science, 2,1978

Burton, R.R, Brown, J.S. "An investigation of computer coaching for informal learning activities" in Sleeman and Brown, 1982

Burton, R.R. "Diagnosing bugs in a simple procedural skill" in Sleeman and Brown, 1982

Bykat, A. "Solution of finite element equations without assembly." ICSI 468 1,1973

Bykat, A. "Implementation of the finite element method" Univ. of London, UK 1974

Bykat, A. "Automatic generation of triangular grids." International J.Num.Meth.Engng 10(6) 1976

Bykat, A. "A note on an element ordering scheme." International J. Num.Meth.Engng 11(1)1977

Bykat, A. "Design of a recursive shape controlling mesh generator." International J.Num.Meth.Engng, 19(9)1983

Bykat, A. "Designing an intelligent operating system consultant and teacher" Proc.IEEE-PCCC-86, pp.572-578 3,1986

Clancey, W.J. "Classification problem solving", AAAI-84, Austin, 1984

Minsky, M. "A framework for representing knowledge" in P.Winston, The psychology of computer vision", McGraw-Hill, 1975

Schank R., Abelson R. "Scripts, Plans, Goals and understanding.", LEA, 1977

Sleeman D., Brown J.S. "Intelligent tutoring systems", Academic Press,1982

Sleeman D. "A self improving quadratic tutor", in Sleeman & Brown, 1982

Weiss, S.M., Kulikowski, C.A. "Designing expert systems", Rowman & Allanheld, 1984

Wilensky, R. "Planning and understanding", Addison-Wesley, 1983

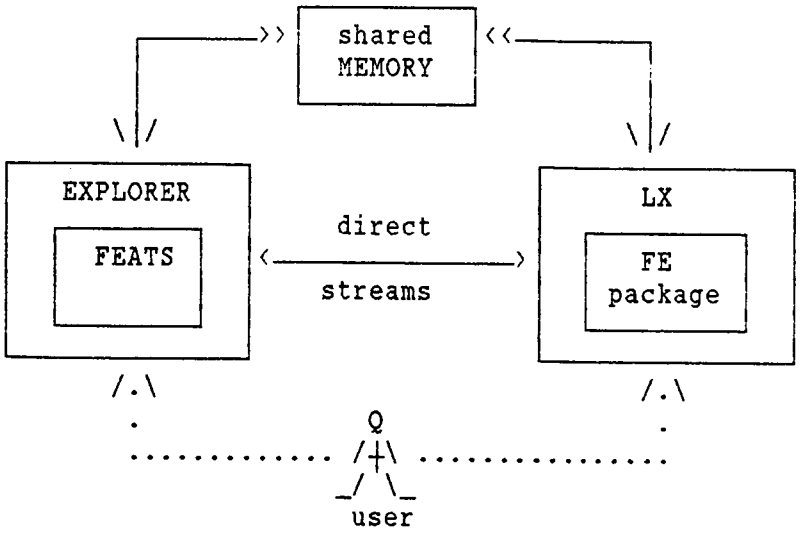


Fig.1 FEATS environment

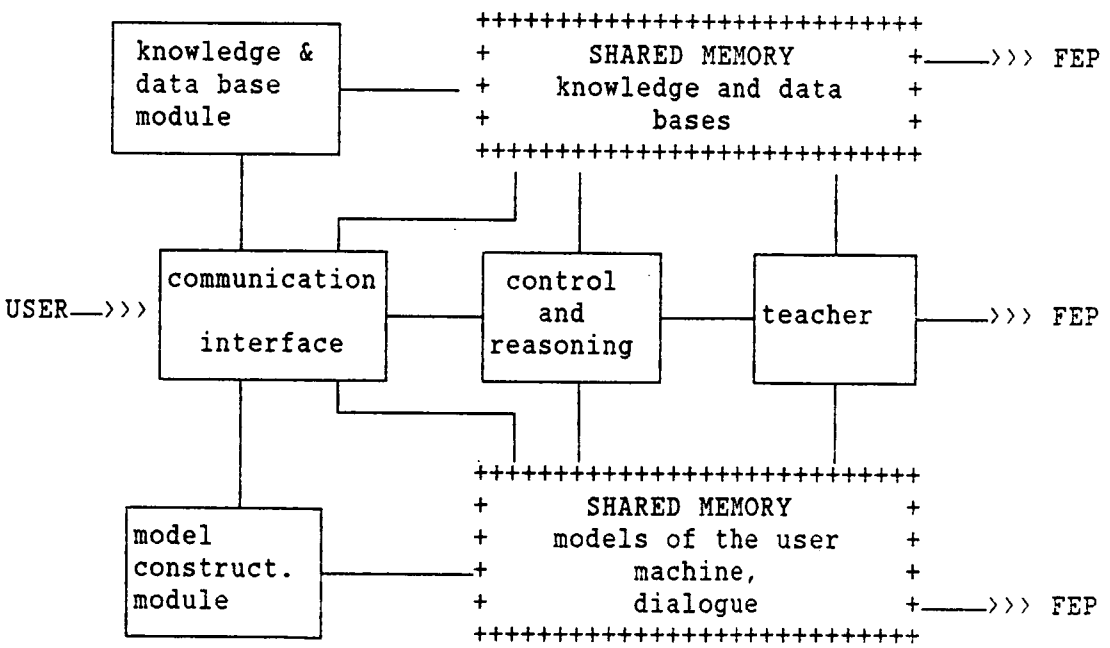


Fig.2 Major components of FEATS

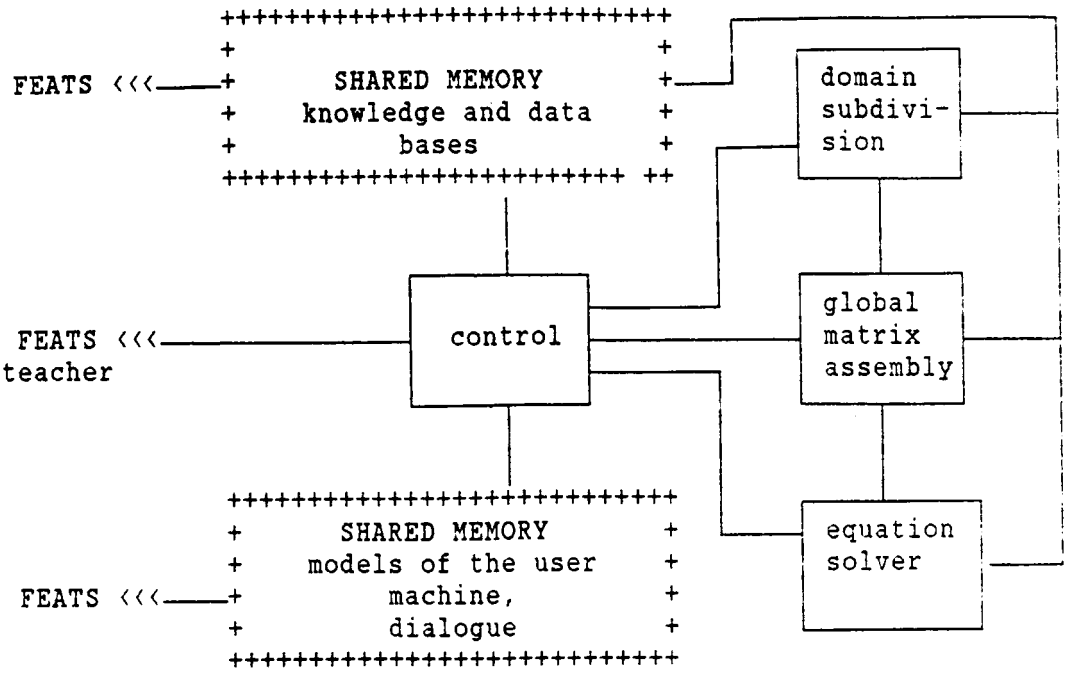


Fig.3 Major components of a finite element package (FEP).

Enhancing your COSMIC NASTRAN usage with PATRAN

Laurie C. Bender and Malcolm P. Johnson
PDA Engineering

SUMMARY

The Mechanical Computer-Aided Engineering (MCAE) market is expanding rapidly through advanced hardware and software systems. The communication channels in the MCAE market is crucial in obtaining the information necessary for design decisions. A design model can be created in one software package, analyzed in several others, and the results processed in still other packages. Loss of data means loss of important design information. PATRAN's neutral file format is designed for storing information in a clear, concise and complete format. This format allows complex design information to be passed between PATRAN and other software packages. PATRAN and COSMIC NASTRAN® are used at many major manufacturing companies worldwide. These companies use the PAT/COSMIC-NASTRAN interface developed by PDA Engineering to transfer design data between these two powerful software programs. The data transferred between PATRAN and NASTRAN includes finite element data, loads and boundary conditions, material and property definitions, and analysis results data. The coupling of these two codes helps an analyst make intelligent decisions regarding his complex design. PATRAN and NASTRAN were used at Deutsch Metal Components for analysis of swage head tooling. Through the use of these two software codes they were able to make important design modifications contributing to increased functionality of the tool.

INTRODUCTION

MCAE is the process of defining a physical model of a design in a computer, then subjecting that model to a simulated environment to determine its response (ref 1). By analyzing the model's reaction to the applied loads, the design can be verified and optimized.

The great benefit of MCAE is that it allows the engineer to take his design from conception to reality with less need for prototypes. More "what if" questions can be asked by the engineer, improving the design, shortening the development cycle and reducing the product costs.

Until recently, MCAE has been composed of a range of valuable but incompatible software tools, each designed to perform some aspect of the CAE function–structural analysis, thermal analysis, composite materials analysis, kinematics, and others, plus pre- and post-processing.

The PATRAN System, however, offers an extensive MCAE software interface system. PATRAN not only has the capability to perform many of the MCAE functions itself, it also gives the engineer access to many existing software tools, and makes all those tools easily accessible–and useful. This allows the engineer to model the design, model the

environment, analyze the model within the environment and interpret the results, and then optimize that design.

The PATRAN System is composed of PATRAN Plus, application modules and application interfaces. The gateway utilities of PATRAN allow users the freedom to define their own software environment, while application interfaces bring the "world" of MCAE into PATRAN. Virtually every major finite element code such as NASTRAN, as well as many important computer aided drafting and manufacturing software packages, can be tightly linked into the PATRAN interface system.

The PATRAN System

PATRAN is an open-ended, general purpose, 3-D mechanical computer aided engineering (MCAE) software package that uses interactive graphics to link engineering design analysis and results evaluation functions. The package includes an advanced solid modeler, extensive graphics imaging capabilities, the industry's acknowledged leading finite element modeler, interactive representation of analysis results, and a unique, open-ended "gateway" architecture that facilitates access to virtually every design, analysis and manufacturing software program.

PATRAN provides users with the ability to conceptualize, develop, and test a product on the computer prior to committing manufacturing and material costs. Its powerful yet concise command structure permits realistic, detailed model representations to be generated on most major hardware configurations, from workstations to super computers. The package consists of five tightly integrated modules, including P/SOLID, P/FEM, P/IMAGE, P/POST, and P/PLOT, plus G/GATEWAY.

P/SOLID is a geometric modeling system that incorporates both analytic solid modeling (ASM) and trimmed surface modeling (TSM) techniques. ASM defines entities based on parametric cubic curves, surfaces, and solids. TSM represents bodies by their surfaces, a collection of trimmed bicubic surface patches. For solid model generation using Boolean operations, TSM is optimal. ASM permits mass property calculations, including fixed or variable properties such as density. ASM also provides the link to finite element mesh generation for two and three dimensional objects, and spatially dependent boundary conditions. The two modeling methods are interwoven, allowing the engineer to use both simultaneously and interactively. P/SOLID's integration into PATRAN Plus makes it easy to accurately conceptualize, model, and modify potential designs.

P/FEM helps prepare models for analysis. The geometry created by P/SOLID is accessed directly to develop a finite element mesh, apply loading and boundary conditions, and define physical properties. Because of the strong tie between P/FEM and P/SOLID, a finite element mesh is easily developed from the geometric model, permitting generation of multiple code-specific meshes and constraints. Meshes can be uniform across a model or concentrated around critical regions, supplying the needed refinement to examine design concerns. P/FEM provides capabilities to help insure the integrity of the mesh, including plate element checking. Additionally, the module uses P/IMAGE to display and verify all data prior to executing an analysis.

P/IMAGE encompasses the complete graphics capability found within PATRAN Plus. The module includes graphic feedback for all commands, provides presentation shading, and serves as a visual verification prior to executing an analysis. P/IMAGE features a number of options that take advantage of the hardware's capabilities, including local view

manipulation, local shading, multiple light sources, and transparency. Apart from these added enhancements, the PATRAN display is similar across all machines. This makes it easy for the user to learn the program within a heterogeneous hardware environment.

P/POST quickly and clearly displays analysis results onto a PATRAN model. Results can be structural, thermal, fluid, magnetic, or any other application where the resultant values are associated with their respective nodes or elements. P/POST eliminates the need for stacks of printout, making it easy for the user to understand the analysis results and determine critical regions. Its tight integration into the PATRAN Plus package allows users to super-impose results directly onto a P/FEM model, and subsequently modify the design according to optimization requirements. In-house codes should have no trouble interfacing to the post-processing file's simple format. P/POST employs a variety of means to depict results, including animation, deformed geometry plots, contour plots, fringe plots, carpet plots, vector plots, and X-Y plots for beam elements.

P/PLOT, the newest module to be nested within PATRAN Plus, generates engineering X-Y plots. The module permits the user to display and compare two generic data sets, results vs. location for example, and assists in evaluating a design. Its coupling to the other modules of PATRAN Plus enables the user to easily generate multiple graphs from within the PATRAN system environment.

Gateway Utilities

G/GATEWAY constitutes a collection of utility programs and features that enable a user to join PATRAN Plus with external software packages. Utilities supplied with G/GATEWAY allow easy data transfer with PATRAN Plus, providing the link needed to interface between different software packages. G/GATEWAY permits PATRAN Plus to run on a variety of hardware configurations, giving users a wide choice of operating environments. Other features include a number of separate utility programs to assist in the documentation, presentation, and manipulation of the information output by PATRAN Plus, as well as other application software.

PATRAN's open architecture can be used in a variety of ways for the exchange of useful information. G/GATEWAY features have broad implimentations across software and hardware systems. There are literally hundreds of ways in which the user can interact with PATRAN files. Below are just some examples of the use of G/GATEWAY which are provided in the standard PATRAN package. Later we will concentrate on the PATRAN neutral file, which is the file used to communicate with NASTRAN.

G/DB-Access (DBXS) is a collection of FORTRAN utilities that access the PATRAN Plus database directly. It permits other applications to directly read PATRAN data through the GATEWAY system.

PATRANIFC permits users to customize the interface menu. It includes calls to PATRAN System supported Application Interfaces but can include invocation of any external software package.

HARDCOPY is a program to reformat PATRAN Plus generated graphic files into commands.for CALCOMP and compatible plotters, For TRILOG and PRINTRONIX dot matrix printers on some computer systems, and has ancillary support for the TEKTRONIX 4510 rasterizer.

OPTIONSET automatically customizes the PATRAN session environment through a simple command file.

The Neutral System is the most commonly used method of linking PATRAN to other software packages. The neutral file is a text formatted or binary file, generated and/or read by PATRAN, which contains selected PATRAN information. The neutral file includes geometric model data, finite element definitions and associated properties, loads and boundary conditions, and groupings of entities called Named Components. The neutral file is used to communicate with analysis codes such as NASTRAN.

PAT/COSMIC

PAT/COSMIC is the link between PATRAN's pre- and post-processing and NASTRAN's analysis of a model (ref 2). It consists of two software programs: PATCOS, which takes PATRAN neutral file data and converts it to a NASTRAN bulk data deck for analysis; and COSPAT, which converts a NASTRAN OUTPUT2 file into PATRAN-compatible results files. COSPAT can also be used to read a NASTRAN bulk data deck and convert it into a PATRAN neutral file. PAT/COSMIC is an interactive program. Inputs required from the user are minimal and execution time is short.

PATCOS can produce 59 different NASTRAN bulk data cards, including 29 different element types. Prompts and other aids built into the program should enable a new user to obtain a successful PATCOS run on the very first try, without any external instructions.

If desired, three parameters may be set during PATCOS execution: MINSD, LGRID, and APZERO. MINSD defines the minimum permissible number of significant digits for real values on the bulk data cards. LGRID determines which coordinate frame is specified on the GRID card, whether the frame used during creation of a node or the global coordinate frame. APZERO is a value specified which causes all values less than that to be set to absolute zero during translation (i.e. if APZERO is set to 1.0E-4 and a node is identified as having a coordinate value of 1.0E-5 in PATRAN, PATCOS will set that coordinate in the GRID card to 0.0).

The complete list of supported card types are contained in Table 1.

COSPAT creates PATRAN-readable results files from a NASTRAN OUTPUT2 file. These results files include nodal displacements, element centroidal stresses and strains, and nodal stresses. In order to generate an OUTPUT2 file from a NASTRAN analysis, DMAP Alter sequences must be included in the bulk data deck prior to analysis. DMAP Alter sequences are provided with COSPAT for the most commonly used solution sequences.

COSPAT results files can be read into PATRAN for post-processing. The results files are in a column format, with various columns of data associated with each node or element. For example, first principal stresses are contained in column 22 for CHEXA1 elements.

As mentioned previously, COSPAT can also read a NASTRAN bulk data deck and create a PATRAN neutral file. This could be very useful for a new PATRAN user who already has NASTRAN models stored on his computer. Many companies have taken finite element models which were not built with a graphics pre-processor and brought these models into PATRAN. The users were surprised to find errors in their modeling technique, such as "bow-tied" elements (incorrect connectivity turns rectangular shaped elements into a bow-tied shape), that are only apparent with graphics systems. By utilizing PATRAN these

companies were able to go back and correct modeling errors and design judgements based on these incorrect analyses.

Also, many large companies provide sub-contractors with already-built finite element models for further analysis. These NASTRAN models can be easily read into PATRAN for modification, such as changes in the design or material properties. After the changes are made, the analyst can simply write out another neutral file, run PATCOS to create a NASTRAN bulk data deck, and take the new model into NASTRAN for further analysis.

Deutsch Metal Components is a good example of how a company uses PATRAN and NASTRAN to design and analyze a specific part.

PATRAN and NASTRAN at Deutsch Metal Components

Deutsch Metal Components is a manufacturer of Permaswage® advanced tube connecting systems and swage tooling for the Aerospace, Marine, and Oil industries. Deutsch has one of the largest manufacturing facilities in Southern California. Advanced equipment at Deutsch includes computerized order processing and inventory control, CNC manufacturing, and the latest in CAD/CAM systems.

Deutsch currently has three PATRAN users running on a PRIME 2655 computer. PATRAN has been in-house at Deutsch since February 1985. Initial designs are created using PRIME MEDUSA and translated to PATRAN via an interface supplied by PRIME. The finite element models, including nodes and elements, material and property definitions and loading conditions are created in PATRAN. Hardcopies of the finite element model and analysis results were obtained by PATRAN through a Tektronix 4115 terminal hooked to a Tektronix 4692 ink-jet plotter. Finite element data is passed to NASTRAN for analysis via PAT/COSMIC.

Deutsch initiated a redesign of their swage tooling, which radially compresses fittings onto pipes, eliminating costly welding of these pipes. A hydraulic power unit is connected to the swage head tooling and provides the force needed to fit the pipes together. Prime consideration in the redesign of the swage tooling was reduction ot the swage head radius. This radius controls the distance between two piping systems. The smaller the swage head, the closer together the pipes can be placed, creating a more efficient piping system environment. Other design considerations of the swage head were weight and cost of manufacturing. PATRAN and NASTRAN were used to minimize the swage head radius while keeping the stress levels generated in the part under the maximum allowable stress levels.

The swage tooling is comprised of three parts: the swage head, the die block, and the cylinder (see Figure 1). The finite element model was created with 2D axisymmetric elements. Vertical force loadings were applied to the model to simulate the hydraulic pressure translated through the swage head. Single point constraints were applied along the vertical axis as well as axisymmetric boundary conditions. PATCOS created the NASTRAN bulk data deck from the PATRAN model. The analysis took 1-2 hours on the Prime computer. Results were translated back into PATRAN for post-processing.

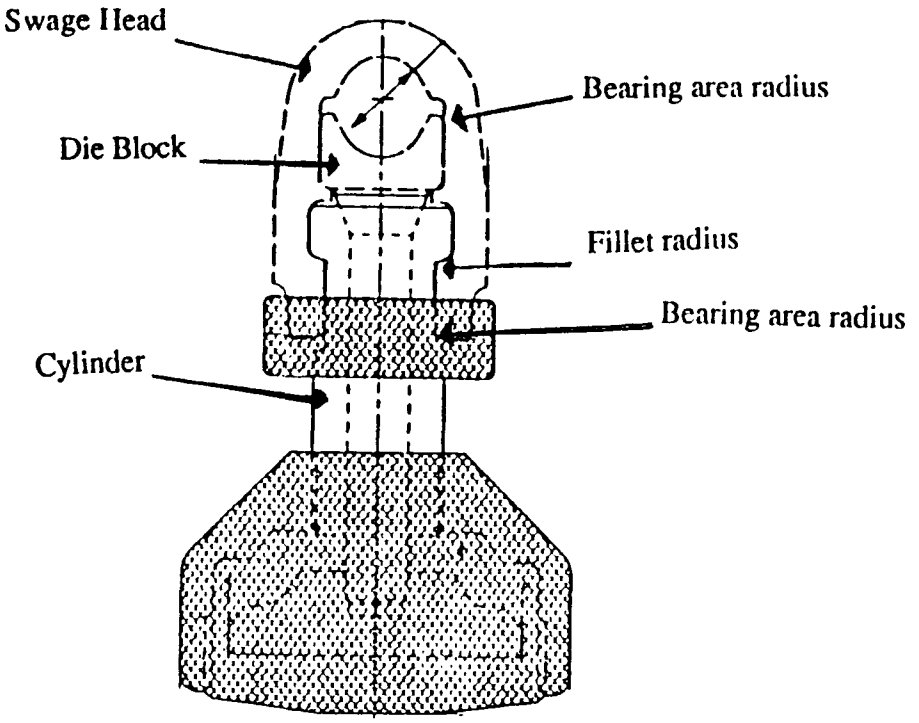


Figure 1. Initial swage head tooling design

After approximately 6-8 design iterations, a final configuration was developed. This configuration was translated across the entire line of Deutsch swage head tooling. Design changes made to the tooling consisted of modification of the bearing head radii and fillet radius. Also, the material of the tooling was changed from 300 maraging steel to PH13-8M stainless steel. The stainless steel has a higher maximum stress than maraging steel for 100,000 fatigue life cycles, the design criteria for the tooling. This final design reduced the swage head radius by approximately 30%, and reduced the weight of the tooling by approximately 83%. Also, the nut which held the swaging head to the hydraulic power unit was eliminated in the final design (see Figure 2).

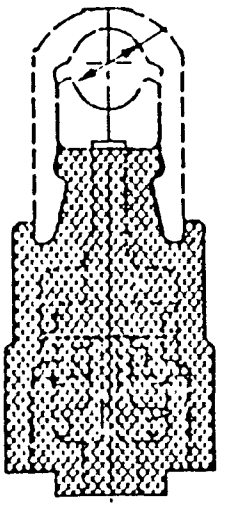


Figure 2. Final design configuration of Swage head tooling

REFERENCES

1. PATRAN Plus User Manual. PDA Engineering, July 1987, pp. 1-9 - 1-24.

2. PAT/COSMIC-NASTRAN Application Interface Guide. PDA Engineering, October 1984.

TABLE 1

COSMIC/NASTRAN Card Types Supported by PATCOS

Element Definitions

CBAR
CELAS2
CHBDY
CHEXA1
CHEXA2
CIHEX1
CIHEX2
CIHEX3
CONM2
CQDMEM1
CQDMEM2
CQDPLT
CQUAD1
CQUAD2
CROD
CSHEAR
CTETRA
CTRAPRG
CTRBSC
CTRIA1
CTRIA2
CTRIARG
CTRIM6
CTRMEM
CTRPLT
CTRSHL
CWEDGE
CNGRNT

Element Properties

PBAR
PHBDY
PIHEX
PQDMEM1
PQDMEM2
PQDPLT
PQUAD1
PQUAD2
PROD
PSHEAR
PTRBSC
PRTIA1
PTRIA2
PTRIM6
PTRMEM
PTRPLT
PTRSHL

Node Coordinates

GRID

Coordinate Frames

CORD2C
CORD2R
CORD2S

Material Properties

MAT1
MAT2
MAT3
MAT4
MAT5

Node Forces

FORCE
MOMENT

Node Displacements

SPC

Constraints

SPC1

Temperatures

TEMP

Bar Deformation

DEFORM

EXPERIENCES WITH THE QUAD4 ELEMENT FOR SHELL VIBRATIONS

by

Melvyn S. Marcus, Gordon C. Everstine, and Myles M. Hurwitz
Applied Mathematics Division (184)
David Taylor Research Center
Bethesda, Maryland 20084 U.S.A.

ABSTRACT

A new bending and membrane element, the QUAD4, was added to the 1987 release of NASTRAN. The results of a series of evaluations for statics applications were presented by Victoria Tischler of the Wright-Patterson Air Force Base at the 1987 NASTRAN Users' Colloquium. Here we show the results of a QUAD4 evaluation involving the calculation of the natural frequencies of a thin-walled cylindrical shell with flat end caps. The QUAD4 results are obtained using both lumped and coupled mass formulations and compared to results obtained using the conical shell element (with lumped mass), the QUAD2 element (with both lumped and coupled mass), and an ad hoc element which superposes the QDPLT and QDMEM1 elements. For this problem, it is concluded that QUAD4 performs very well if the lumped mass formulation is used. However, with the coupled mass formulation, the QUAD4 performs poorly.

INTRODUCTION

One of the long-awaited enhancements in the 1987 release of NASTRAN was the addition of the QUAD4 element, a four-node bilinear isoparametric membrane-bending element. This element, which was developed for the Wright-Patterson Air Force Base, can handle variable element thickness and layered composite construction. At the 1987 NASTRAN Users' Colloquium in Kansas City, Victoria Tischler of Wright-Patterson presented the results for an extensive set of test problems, all of which involved statics applications.

Since we have particular interest in structural dynamics, we performed a set of calculations to evaluate the QUAD4 element for use in dynamics. The test problem used for this evaluation was the calculation of the natural vibration frequencies and corresponding mode shapes of a thin-walled cylindrical shell with flat end caps.

It was deemed useful to test the QUAD4 using both its lumped and consistent mass formulations, and to compare the QUAD4 with its competition. For general homogeneous shells, the QUAD4's principal competitors are the QUAD2 element and an ad hoc element obtained by superposing the QDPLT and QDMEM1 elements. This latter "element" is often used as a replacement for QUAD2 since it has a better membrane formulation than that used in QUAD2. In

addition, since the test problem does not have an "exact" solution, we also computed the natural frequencies of the shell using a very fine mesh of conical shell (CONEAX) elements. If the individual elements are correctly formulated and coded, all these approaches would presumably converge to the "correct" results (although at different rates). Thus, since the conical shell model is exact in the circumferential direction, a fine mesh of these elements can be used as a benchmark for comparison.

THE TEST PROBLEM

The test shell is a freely-supported thin-walled cylindrical shell with flat end caps, as shown in Fig. 1. Both the shell and the flanges (used to support the end caps) are made of aluminum, for which the assumed material properties were Young's modulus $E = 10.3 \times 10^6$ psi, Poisson's ratio $\nu = 0.33$, and mass density $\rho = 2.524 \times 10^{-4}$ lb-sec^2/in^4. The flat end closures are made of a general purpose grade of Lexan® polycarbonate sheet (made by General Electric), for which the assumed material properties were shear modulus $G = 11.4 \times 10^4$ psi, $\nu = 0.37$, and $\rho = 1.121 \times 10^{-4}$ lb-sec^2/in^4.

THE FINITE ELEMENT MODELS

Six different finite element models were used to compute the natural frequencies of the test shell:

1 - conical shell (CONEAX) elements, lumped mass, 192 elements lengthwise and 17 elements radially on end plate (4438 DOF),

2 - superposition of QDPLT and QDMEM1 elements, lumped mass, 72 elements lengthwise, 24 elements circumferentially, 5 elements radially on end plate, and BAR elements for flange (2970 DOF) (Fig. 2),

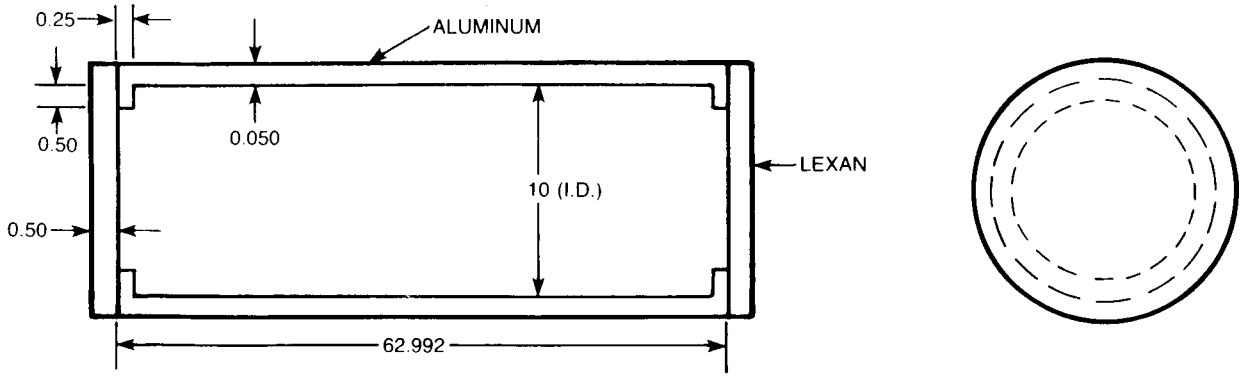


Fig. 1. Cylindrical Shell with Flat End Plates.

3 - QUAD2 elements, lumped mass, same mesh as Case 2,

4 - QUAD2 elements, coupled mass, same mesh as Case 2,

5 - QUAD4 elements, lumped mass, same mesh as Case 2, and

6 - QUAD4 elements, coupled mass, same mesh as Case 2.

In all cases, a single plane of symmetry was imposed at the mid-length, and only the modes symmetric with respect to the mid-length plane were computed. (Thus, only the modes with an odd number of longitudinal half-waves would be found.) For Cases 2-6, half the circumference was modeled, and symmetry boundary conditions were imposed at all points in that symmetry plane. (Since all shell modes have an even number of circumferential half-waves, there are no additional modes which could be found by instead imposing anti-symmetric boundary conditions.) The numbers of elements listed above for the meshes would be the numbers which would have been used if the complete shell had been modeled rather than only half the shell as in Case 1, and one-quarter the shell as in the other cases. Also, for simplicity in modeling, the flanges were assumed to coincide with, rather than be offset (longitudinally) from, the end plates. The flange was modeled with two conical shell elements in Case 1 and with BAR elements (offset radially) in the other cases. The conical shell mesh was prescribed to be much finer than the other meshes so that this model could serve as a benchmark to which the other solutions could be compared.

PRESENTATION OF RESULTS AND DISCUSSION

The first 20 natural frequencies and mode shapes were found for the six finite element models of the cylindrical shell. For all six cases, the eigenvalues were extracted using NASTRAN's FEER method. The results of these calculations are shown in the table on the next page. The second column in the table (Harm. n) denotes the circumferential harmonic index, the number of

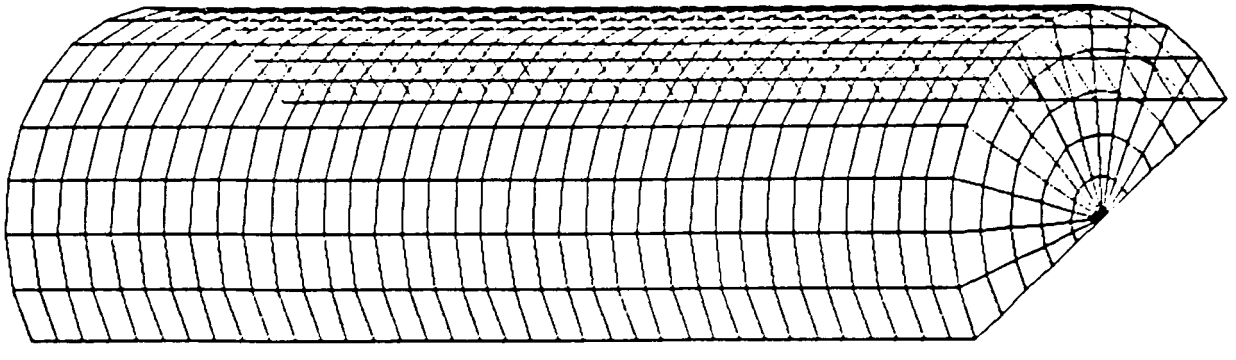


Fig. 2. Finite Element Mesh used for Cases 2-6.

Table. Natural Frequencies of Cylindrical Shell with Flat End Plates

	Mode			Frequency (Hz)					
	Harm.	Shell	Plate	CONEAX	PLT/MEM1	QUAD2	QUAD4	QUAD2	QUAD4
No.	n	m	m	lumped	lumped	lumped	lumped	coupled	coupled
1				0	0	0	0	0	0
2	2	1		103	104	106	104	106	105
3	3	1		155	159	160	158	159	164
4	4	1		286	296	296	293	295	316
5	0		1	319	305	307	304	309	309
6	4	3		364	372	386	369	384	399
7	1	1	(1)	384	383	386	383	386	384
8	3	3		392	394	413	393	411	408
9	5	1		460	484	484	474	480	537
10	5	3		490	511	519	503	515	571
11	5	5		616	632	671	627	667	714
12	4	5		638	646	693	644	689	697
13	6	1		673	722	722	699	715	842
14	6	3		690	735	739	716	732	864
15	2	3	(1)	691	690	705	690	703	698
16	6	5		752	789	815	778	808	941
17	1		1	790	779	781	770	778	797
18	5	7		864	880	961	878	955	1003
19	6	7		888	917	986	914	979	1109
20	3	5		911	916	956	916	952	954

full waves around the circumference. The third column (Shell m) denotes the number of longitudinal half waves. The fourth column (Plate m) denotes the number of nodal circles (plus one) in the end plate. Most of the first 20 modes are either predominantly shell modes or predominantly end plate modes, as can be seen from the table. In two cases (shown parenthetically in the table), the end plate participates at a noticeable, but secondary, level in the motion.

Because of the fineness of the conical shell mesh, the results for Case 1 are probably the best of the six sets of results. The element formulation is exact in the circumferential direction, and the 192 elements used longitudinally would be more than adequate to represent the highest longitudinal mode, which has only seven longitudinal half waves. Another indication that the conical shell results are the best is that, for all modes except the end plate modes, the natural frequencies obtained are lower than the frequencies obtained in the other five cases. Since natural frequencies computed using consistent formulations converge from above (with mesh refinement), we would expect that, had finer meshes been used in Cases 2-6, lower frequencies would have resulted. Thus, we feel comfortable in treating the conical shell model as the benchmark for this problem. In addition, we have a basis for comparing the various quadrilateral models: namely, that in

ranking two models, the one which yields the lower frequencies is probably the better model.

Several observations can be made about the results in the table:

1. The QUAD2 results are insensitive to the choice of mass modeling (lumped or coupled), although the coupled mass formulation yields slightly better results.

2. The QUAD4 (lumped mass) results are very similar to, but slightly better than, those obtained by the superposition of QDPLT and QDMEM1 elements. This result might be expected, since the membrane part of QUAD4 is the same as QDMEM1 (except perhaps for the number of Gauss integration points used in calculating the stiffness matrix). Both these models are only slightly worse than the conical shell model, even for modes with six circumferential harmonics (n = 6), where the use of only 24 quadrilateral elements in the circumferential direction seems coarse. Both these models are better than the QUAD2 models.

3. The QUAD4 (coupled mass) results are satisfactory only for the lowest few modes. For the higher circumferential harmonics, this element yields results which are in considerable error. An interesting characteristic of the QUAD4 (coupled mass) results in the table is that all four of the n = 6 frequencies exceed the CONEAX results by 25%, and the n = 5 frequencies computed by the QUAD4 (coupled mass) model exceed the CONEAX results by about 17%.

CONCLUSIONS

The QUAD4 element performs well when the lumped mass formulation is used, but poorly when the coupled mass formulation is used. This poor performance is evidently due either to a bad formulation of the mass matrix or to a coding error in the program. Although the latter seems more likely, the issue is as yet unresolved. Until the problem is corrected, we therefore recommend that the coupled mass formulation not be used with the QUAD4 element. As an alternative, we recommend that general shells be modeled either with QUAD4 using a lumped mass formulation or with the superposition of the QDPLT bending element with the QDMEM1 membrane element. The latter approach is probably safer until more is learned about the QUAD4. In any event, both these approaches are preferred over the QUAD2 element.

In general, the evaluation of an element is very difficult, particularly for shells, where one rarely has a theoretical solution to use as a benchmark. Two extensions to this work would be of interest. First, since the element aspect ratio used in our quadrilateral mesh for the cylindrical shell was about 1.5, it would be interesting to repeat the calculations with a unit aspect ratio to see the extent to which aspect ratio is an issue. Second, with such a mesh and with a corrected coupled mass matrix for the QUAD4 element, it would be interesting to extract more modes so that it can be determined whether the coupled mass formulation can be safely used at higher frequencies than can the lumped mass formulation.

COUPLED MASS FOR PRISMATICAL BARS

by

T. G. BUTLER
BUTLER ANALYSES

INTRODUCTION

If one poses the question, "How good is the algorithm, called coupled mass, for apportioning the mass of bars between grid points?", he can get an answer to that question by running a few analytical tests. The classic text in vibrations by Timoshenko[1] provides closed form solutions to the frequency equations for simply supported uniform bars. So the tests that are logical to run involve simply supported bars (hinged) under various combinations of parameters. The results can be checked by substituting the test parameters into the appropriate Timoshenko frequency equation and then by comparing frequencies for corresponding modes. The next question to ask is, "Can the mass coupling algorithm be improved?" One is inclined to think so, because (1) the algorithm in 1987 NASTRAN considers only how the mass is distributed along the length and not how mass is distributed over the cross-section; and (2) it couples this translational mass distribution to the gird points at the end of an element through the static displacement due to bending and ignores displacement contributions from shear.

Last year I presented a paper[2] on this topic and did a lot of right things, but I did one thing wrong which took all the steam out of the paper. This year I will take one step backwards and recoup my goof, then I will advance the topic by including deformation due to shear. The goof that I made in 1987 was to use the wrong theoretical basis for judging the merits of the results. Now when the proper criterion is used, the conclusions are as rewarding as I had hoped that they would be.

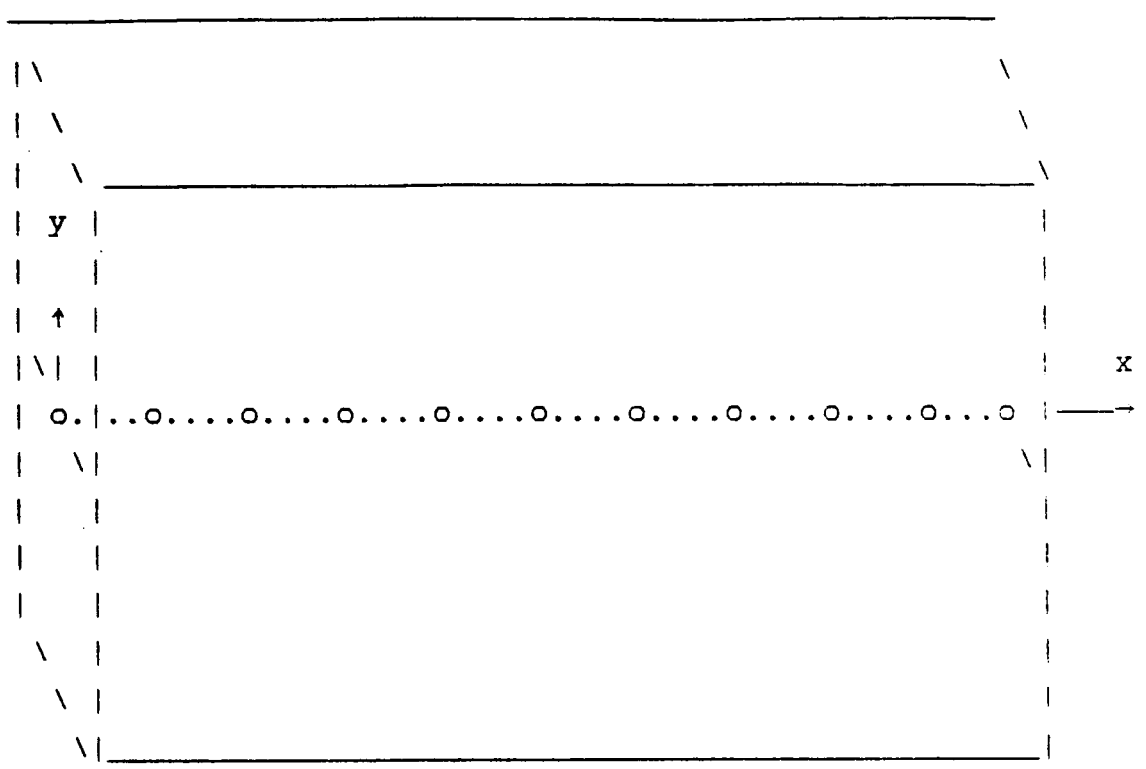
The parameters to be controlled fall into 3 categories: namely geometric, elastic, and mode of mass coupling.

ABSTRACT

Coupled mass for bars in bending has been investigated. The inclusion of rotary inertia for the case in which shear is ignored (infinite) has a beneficial effect. Once shear effects are included, there is some question as to how well static deflections approximate the dynamic shape.

Test Basis

Tests will be run by analyses using NASTRAN. The geometric parameters will be invariant throughout all tests. The bar will be prismatic, 20" long, of rectangular section 4" x 1", with freedoms in bending and none in axial or torsion. The ends will be pinned so as to constrain transverse translations, and allow end rotations about the the 2 transverse axes. The sketch shows the directions of both element and basic coordinates.



Only bending modes are being compared, because the significance of axial modes was brought out in the previous paper and torsional modes will be the topic of a separate paper. Modes will be compared by classes according to bending in the plane of the deep section or in the plane of the shallow section. The beam will be modeled with eleven equally spaced grid points along its length. Thus the solution set will have 2 transverse displacements and associated rotations at each of the 9 interior points and 2 rotational d.o.f.'s at each end point for a total of 40

d.o.f.'s. This will allow for the development of 10 modes of bending about each transverse axis.

The elastic parameters have fixed and variable values. Young's modulus will be fixed at 10×10^6 #/in^2 and the shear modulus will vary amongst the 3 values of 0.0, 3.75×10^6 #/in^2 and ∞.

The inertia parameters have fixed and variable values. The total mass will be held constant. The density will be fixed at 2.588×10^{-4} # sec^2/in^3. Four cases of mass coupling will be formulated:

A. Translational Mass coupled to the end points through the transverse displacements due to static action in bending from unit deformation in end point freedoms.

B. Rotational Inertia coupled to the end points through the slopes of transverse displacements due to static action in bending from unit deformation in end point freedoms.

C. Translational Mass coupled to the end points through the transverse displacements due to static actions in combined bending and shear from unit deformations in end point freedoms.

D. Rotational Inertia coupled to the end points through the slopes of transverse displacements due to the static actions in combined bending and shear from unit deformations in end point freedoms.

Behavior for various combinations of these 4 coupled formulations will be investigated.

Theoretical Basis

References for these tests will be based on the Bernoulli-Euler theory of prismatical beams. This theory is well explained in Stephen Timoshenko's book "Vibration Problems in Engineering" second edition,[1] July 1937, published by Van Nostrand Co., in sections 54 through 58. The solution for a beam with hinged end conditions is given in section 56 pages 338 through 342.

The frequency equation for the most general case which includes rotary inertia and shear is equation 149 of the reference. In that equation are several symbols which will be defined first.

$a^2 = \frac{EI}{\rho A}$ $r^2 = \frac{I}{A}$ $p_n = 2\pi f_n$, where E = Young's modulus

I = area moment of inertia

ρ = mass density

A = area

n = mode number

L = length

k = shear constant

G = shear modulus

f_n = cyclic frequency

$$(149)\quad \frac{a^2 n^4 \pi^4}{L^4} - p_n^2 - p_n^2 \frac{n^2 \pi^2 r^2}{L^2} - p_n^2 \frac{n^2 \pi^2 r^2}{L^2} \frac{E}{k G} + \frac{r^2 \rho}{k G} p_n^4 = 0.$$

This can be particularized by recognizing that certain terms represent individual effects. The presence of k and G in the

last two terms, indicates shear effects. The last three terms contain "r" which entered the derivation from the consideration of rotary inertia.

Case 1

Frequency equation for no shear and no rotary inertia effect. It consists of only the first two terms of equation (149).

$$f_n = \frac{p_n}{2\pi} = \frac{an^2\pi}{2L^2} = \frac{n^2\pi}{2L^2}\sqrt{\frac{EI}{\rho A}}$$

Case 2

If shear is considered infinite and rotary inertia effects are taken into account, the first 3 terms of (149) are non-zero, and the frequency equation reduces to

$$f_n = \frac{n^2\pi}{2L}\sqrt{\frac{EI/\rho A}{L^2 + n^2\pi^2 r^2}}$$

Case 3

If shear is considered without rotary inertia, the identity of contributing terms is much less evident. Appendix A is attached to derive this form of the frequency equation.

$$f_n = \frac{n^2\pi}{2L}\sqrt{\frac{E/\rho}{4n^2\pi^2 + L^2/r^2}}$$

Case 4

To get the expression for frequency explicitly for the general case, treat equation (149) as a quadratic in p_n^2. This is worked out in Appendix A.

$$f_n = \frac{1}{2\pi}\sqrt{\frac{D + \sqrt{DxD - 4CF}}{2F}}$$, where the definitions of C, D, & F are given in Appendix A.

A table is inserted here to indicate the frequency excursions that can occur from strictly a theoretical standpoint, for such parameters as mode number, shear deformation, and rotary inertia effects.

HINGED BAR
THEORETICAL BERNOULLI-EULER
FREQUENCIES CYCLES PER SEC.

MODES IN THE SOFTER DIRECTION

MODE	A. NO SHR NO RTY	A/D	B. INF SHR RTY NRT	B/D	C. SHEAR NO RTY	C/D	D. SHEAR RTY NRT
1	222.84	1.01	222.61	1.00	221.93	1.00	221.70
2	891.35	1.02	887.71	1.00	877.04	1.00	873.68
3	2,005.54	1.04	1,987.23	1.04	1,935.19	1.01	1,919.83
4	3,565.40	1.08	3,508.16	1.06	3,351.68	1.01	3,309.30
5	5,570.93	1.12	5,433.04	1.09	5,073.68	1.02	4,985.67
6	8,022.14	1.16	7,740.76	1.12	7,046.49	1.02	6,894.36
7	10,919.03	1.22	10,407.32	1.16	9,218.36	1.03	8,986.76
8	14,261.59	1.27	13,406.71	1.19	11,543.48	1.03	11,222.00
9	18,049.82	1.33	16,711.72	1.23	13,983.29	1.03	13,566.84
10	22,283.73	1.39	20,294.73	1.27	16,506.61	1.03	15,994.85

MODES IN THE STIFFER DIRECTION

MODE	A. NO SHR NO RTY	A/D	B. INF SHR RTY NRT	B/D	C. SHEAR NO RTY	C/D	D. SHEAR RTY NRT
1	891.35	1.08	877.04	1.06	837.92	1.01	827.32
2	3,565.40	1.27	3,351.68	1.19	2,885.87	1.03	2,805.50
3	8,022.14	1.53	7,046.49	1.34	5,427.86	1.03	5,255.55
4	14,261.59	1.82	11,543.48	1.47	8,092.84	1.03	7,855.99
5	22,283.73	2.12	16,506.61	1.57	10,758.85	1.03	10,489.81
6	32,088.57	2.45	21,711.46	1.66	13,396.53	1.02	13,117.44
7	43,676.10	2.78	27,024.29	1.72	16,003.68	1.02	15,727.13
8	57,045.34	3.11	32,371.36	1.77	18,584.67	1.01	18,317.12
9	72,199.27	3.46	37,714.48	1.81	21,144.58	1.01	20,888.96
10	89,134.91	3.80	43,035.40	1.84	23,687.77	1.01	23,445.07

TABLE 1

Theoretically, inclusion of shear is more important than inclusion of rotary inertia effects. The higher modes are more sensitive to omission of refinements. Without any refinements the 10th mode can have 40% error in the class of soft modes and can have error of a factor of 4 for the class of stiff modes. Just including rotary inertia without shear can cut the error by 60% in the class of soft modes and by a factor of 2 in the class of stiff modes. Just including shear without rotary inertia can keep modes within 3% of accurate values. It remains to be seen how the scheme of consistent mass with and without rotary inertia based on the assumption that static deformation is representative of dynamic behavior for purpose of computing inertia coupling.

RESULTS

Returning to the unfinished business from last year's paper, the wrong impression can be quickly dispelled by comparing logarithmic plots of the ratios of computed to theoretical frequencies. Errors range over 4 decades from .01% to 100% against 10 modes in each class.

Figure 1 represents a model for which the shear modulus can legitimately be neglected and rotary inertia can be ignored. The simplified frequency equation is included in the legend. This is a popular practice of many NASTRAN users today. Standard COUPMASS was used; which means that only translational mass was coupled to the ends through the static delfections from bending without shear. Modes in both the soft and stiff classes fall on top of one another. The first 3 modes contain < 0.1% error. The

first 5 modes contain < 0.5% error. The first 6 modes contain 1% error. The first 9 modes contain < 4% error. Only when the modal harmonic is > 2 less than the total mass points does the error make a sudden jump. This confirms last year's results and the results that Archer[3] published.

Next, Figure 2 makes a proper comparison of models having infinite shear, when one couples only translational inertia and the other couples both translational and rotational inertias through the static deflections from bending only. Note that the frequency equation is like the one in figure 1 without rotary inertia except that L^2 is replaced by $L\sqrt{L^2 + n^2\pi^2r^2}$. This tends to depress the frequency with mode number and with high ratios of I/A. The curves marked BBR1 and BBR2 are plots of the NASTRAN models which used mass coupling of both translational and rotational inertias. Their accuracy picture relative to this set of theoretical frequencies is almost a duplicate of the curves in Figure 1 for the translational coupling case versus its simplified theoretical frequencies. To demonstrate the improvement of adding rotary inertia, the ratios of modal frequencies of the first case, without rotary inertia, to the second frequency equation are drawn on this plot. That marked BBT1 is for the stiff class of modes acting in plane 1 and BBT2 is for the soft class of modes acting in plane 2. Note that none of the modes of BBT1 have errors less than 5% and accumulate 100% error as it reaches the 9th mode. The soft class, BBT2, produces 6 modes with errors under 5%. In contrast, all 9 modes of both BBR1 and BBR2 are < 5%. The work of the 1987 paper is vindicated!

A lot of work was involved in computing the coupling of translational inertias and then rotational inertias through the static deflections due to both bending and shear. Once calculated, there opens up many permutations of actions. Take the case of the model that includes elastic actions from bending and shear, but no rotary inertia coupling. The translational mass coupling can now take the options of: coupling from displacements acting through only; or coupling from displacements acting through both bending and shear. Results from these two options are displayed in Figure 2 for the two class of stiff and soft modes. They are marked SST1 and SST2 for translational inertia coupled through static displacements from bending and shear; and SBT1 and SBT2 for translational inertia coupled through static displacements from bending only. Note that the frequency equation for this case including shear without rotary inertia is like that for the rotary inertia case without shear except that the term involving r^2 is scaled by the ratio E/(kG). This implies that the frequency is depressed as the shear factor for the cross section "k" increases. But "k" is a non-dimensional shape factor independent of size so "k" is the same for both the stiff and the soft classes. Because "E" is always > "G" the effect of the scaling is to shift importance away from the length and increase the emphasis on mode number and section ration I/A. One would tend to develop a bias towards what results to expect based on the trend of the first 2 cases; i.e. the case of coupling that embraces the more complete set of options in an instance would be expected to perform best. It turns out that the most accurate case is for SBT; i.e. mass which is coupled through displacements due to bending only without shear. All 9 modes for the soft

class SBT2 have errors < 5% and all 9 modes for the stiff class SBT1 have errors < 7%. But the behavior of models with coupling through static displacements due to both bending and shear have more error. The soft class SST2 has the first 4 modes with error < 1%; the first 6 modes with error < 5%; the first 8 modes with error < 10%. For the stiff class SST1 only the first 2 modes have error ratios < 1%; the first 4 modes have error ratios < 5%; the first 5 modes have error ratios < 10%; and the 9th mode rises to an error ratio of 20%. It would appear that for the case of bending with shear, NASTRAN is equipped to handle that well as it now stands by calling for COUPMASS and entering a shear coefficient on the PBAR card. Refining the coupling to embrace the deformations from shear impair instead of benefit the modeling in this case. It is now instructive to see how much error results from using COUPMASS without shear for the case when shear is important. Soft case BBT2 has only the first mode with an error ratio < 1% and the 9th mode climbs to an error ratio of 35%. Stiff case BBT1 has no modes with an error ratio < 6% and the upper modes climb to an error ratio of 350%. BBR1 and BBR2 with rotary inertia are only slightly better than BBT1 and BBT2.

Finally we consider the fully refined case in Figure 4. Elasticity includes both bending and shear. There are 4 permutations of mass coupling to consider: 1. TTR=Translational inertia coupled through static bending displacement only; rotational inertia coupled through static bending displacement only. 2. TSR=Translational inertia coupled through static bending dis-

placement only; rotational inertia coupled through static displacement from both bending and shear.

3. SBR=Translational inertia coupled through static displacement from both bending and shear; rotational inertia coupled through static bending displacement only. 4. SSR=Translational inertia coupled through static displacement from both bending and shear; rotational inertia coupled through static displacement from both bending and shear.

Looking at Table 1 and noting the theoretical ratios of shear with and without rotary inertia, one is inclined to think that the trend obtained for the third case would persist for this fourth case too; i.e. the fully coupled scheme might not be the most accurate. But the plotting of this fourth case will be split in two between responses in plane 1 and plane 2, because there is so much to put on one chart. The plot of plane 2 will continue as a semi-log plot, but plane 1 will be plotted as Cartesian.

The soft class will be considered first. The mass coupling that produced the least error was that for which both translational and rotational inertias were coupled through the displacements from bending only, TTR2. All modes had frequency error ratios < 2%. Only a fraction more and still within 2% were the models using translational inertias coupled through static displacements from bending only while rotational inertias were coupled through static displacements from both bending and shear, TSR2.

When the translational mass is coupled through static displacement from both bending and shear and the rotary inertia is coupled through alternates of with and without shear, SSR2 and SBR2, the error ratio is within 5% for the first 8 modes for both kinds of coupling. For the 9th mode there is only 1% spread between SBR and SSR, so other than this they are almost congruent. Even when no rotational inertia is included, the translational mass coupled through static displacements from bending only, SBT2, maintains the error < 10% for the first 9 modes. When translational mass is coupled through static displacements from both bending and shear, SST2, the error is < 10% for the first 7 modes.

The picture changes dramatically for the class of stiff modes. Only the fundamentals of all 4 types of coupling are within 1% accuracy. Only the 2nd mode for all 4 types is within 3% accuracy. Only the first 3 modes for 4 types have error ratios within 10%. The error increases steeply to 40% in the 8th mode. The one case that has error less than 10% for all nine modes is SST1. This also was best for the case examined in Figure 3.

The general trend from all four of these figures is that coupling of translational mass serves a majority of cases, but rotational inertia is needed for the beams without shear deformation and whose section has sizeable moments of inertia. Rotational inertia is also good for the soft case of the most refined beam. This leads to a possibility that not all bugs were found in doing this work for the rotational inertia stiff class of the

most refined case. The fact that the error is negative suggests and anomaly.

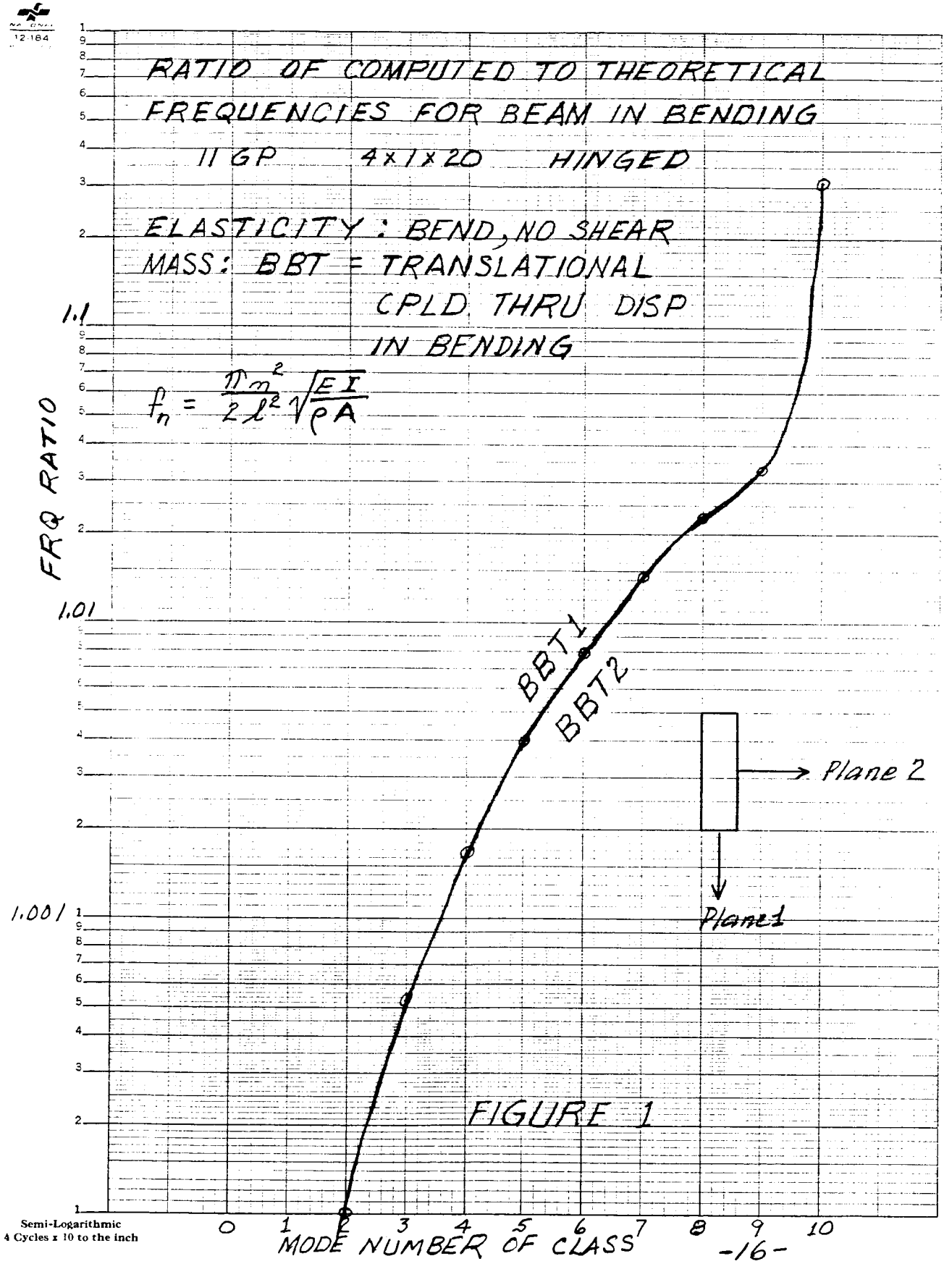
The rule has been confirmed that the highest mode that can be trusted is the one that is 2 less than the number of mass points.

References:

1. S. Timoshenko, "VIBRATION PROBLEMS IN ENGINEERING" D. Van Nostrand Company, Inc.

2. T. G. Butler, "MASS MODELING FOR BARS", Proceedings of the Fifteenth NASTRAN User's Colloquium. NASA Conference Publication 2481. May, 1987.

3. John S. Archer, "CONSISTENT MASS MATRIX FOR DISTRIBUTED MASS SYSTEMS". Journal of Structural Division, Proceedings of the American Society of Civil Engineers, Vol 89, No. ST4, August 1963



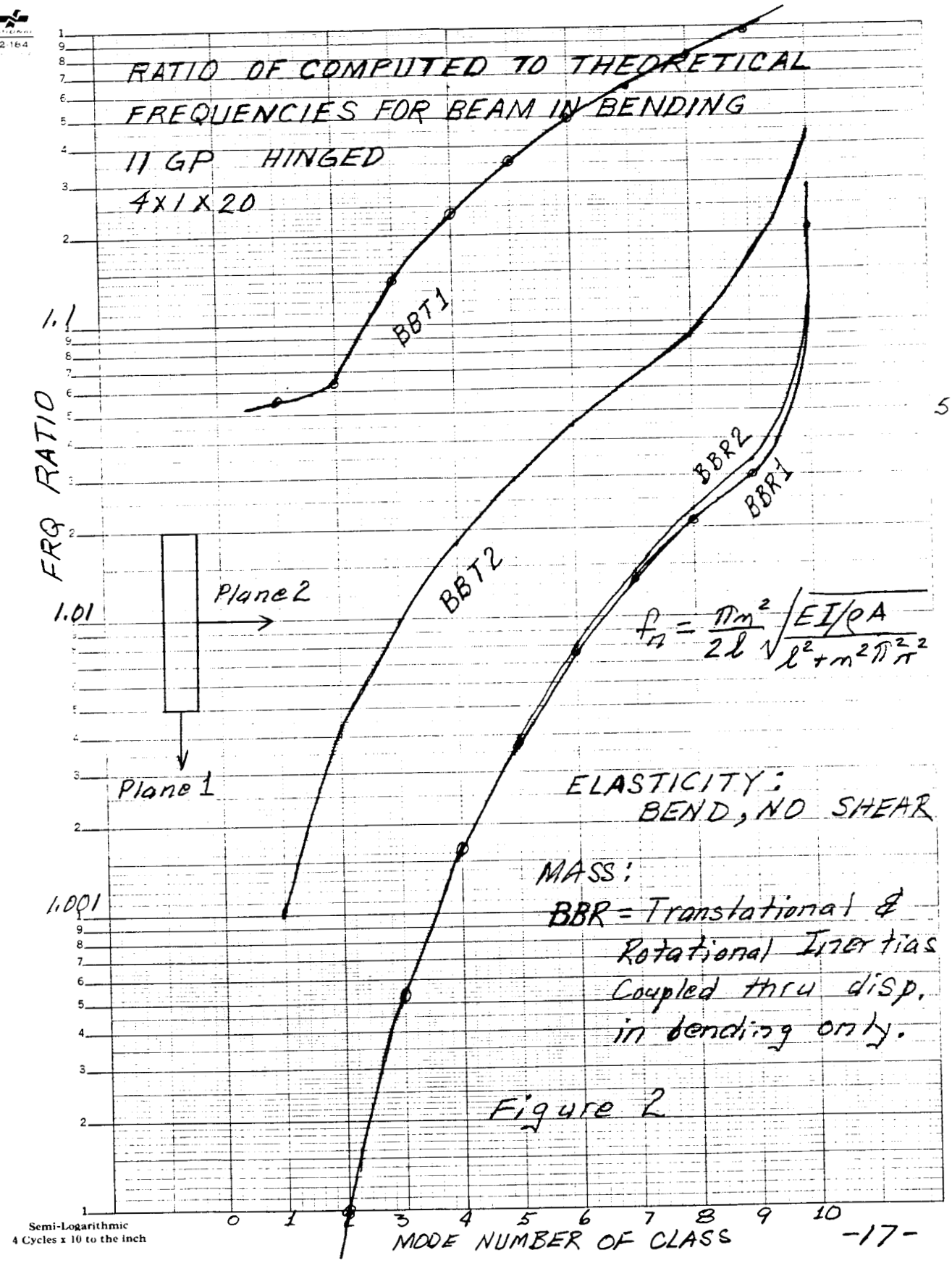


Figure 2

Semi-Logarithmic
4 Cycles x 10 to the inch

-17-

ORIGINAL PAGE IS
OF POOR QUALITY

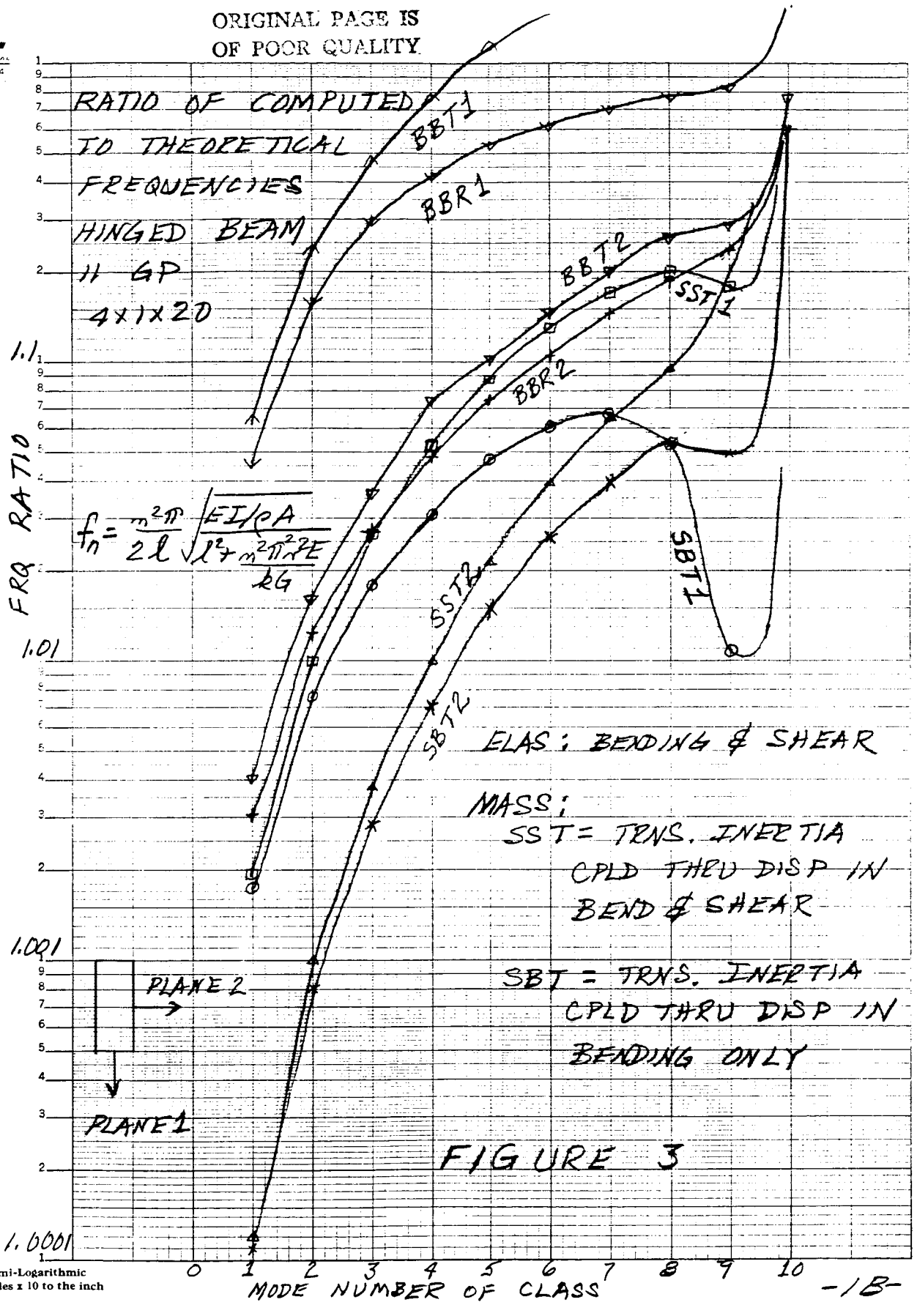


FIGURE 3

-18-

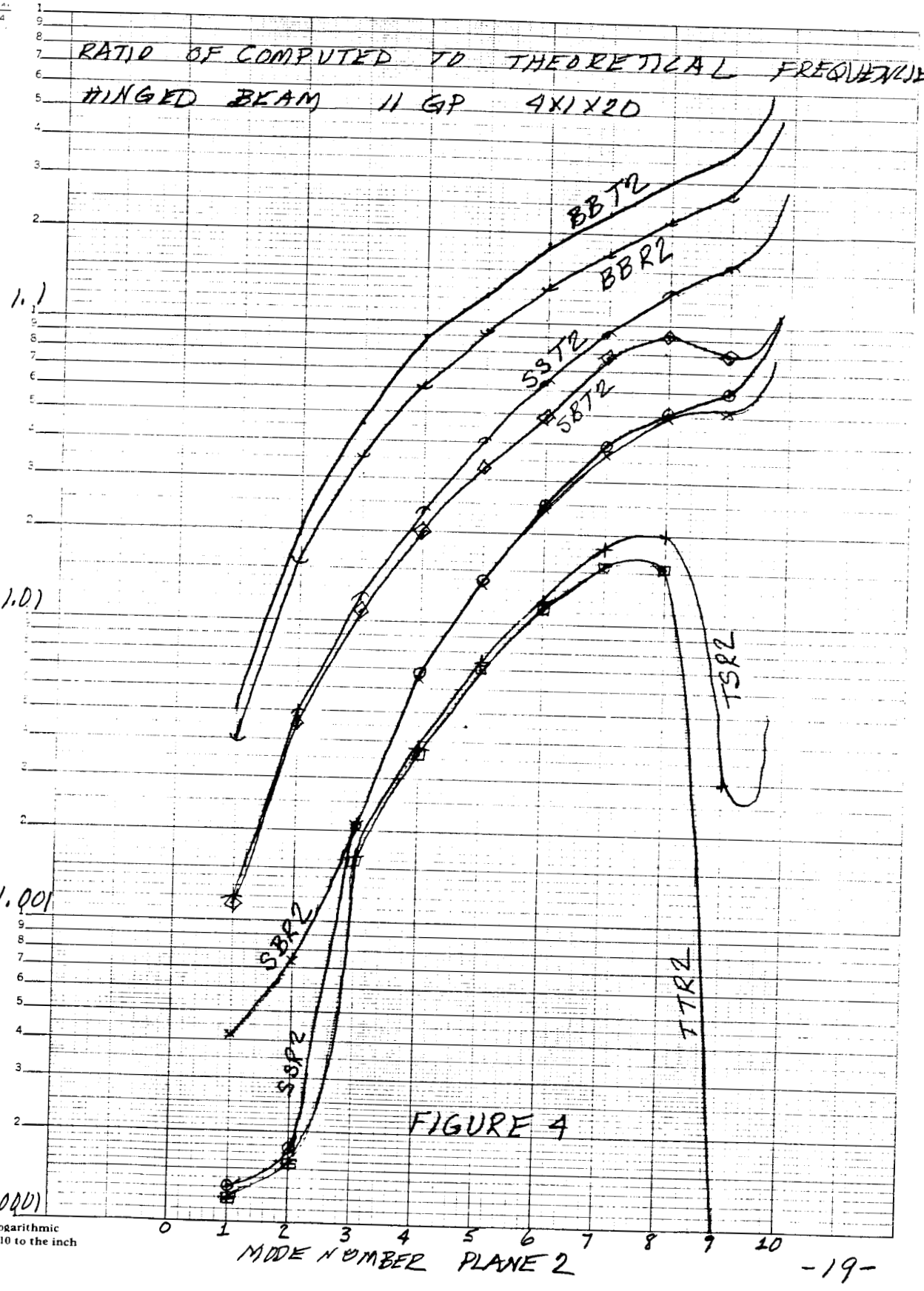
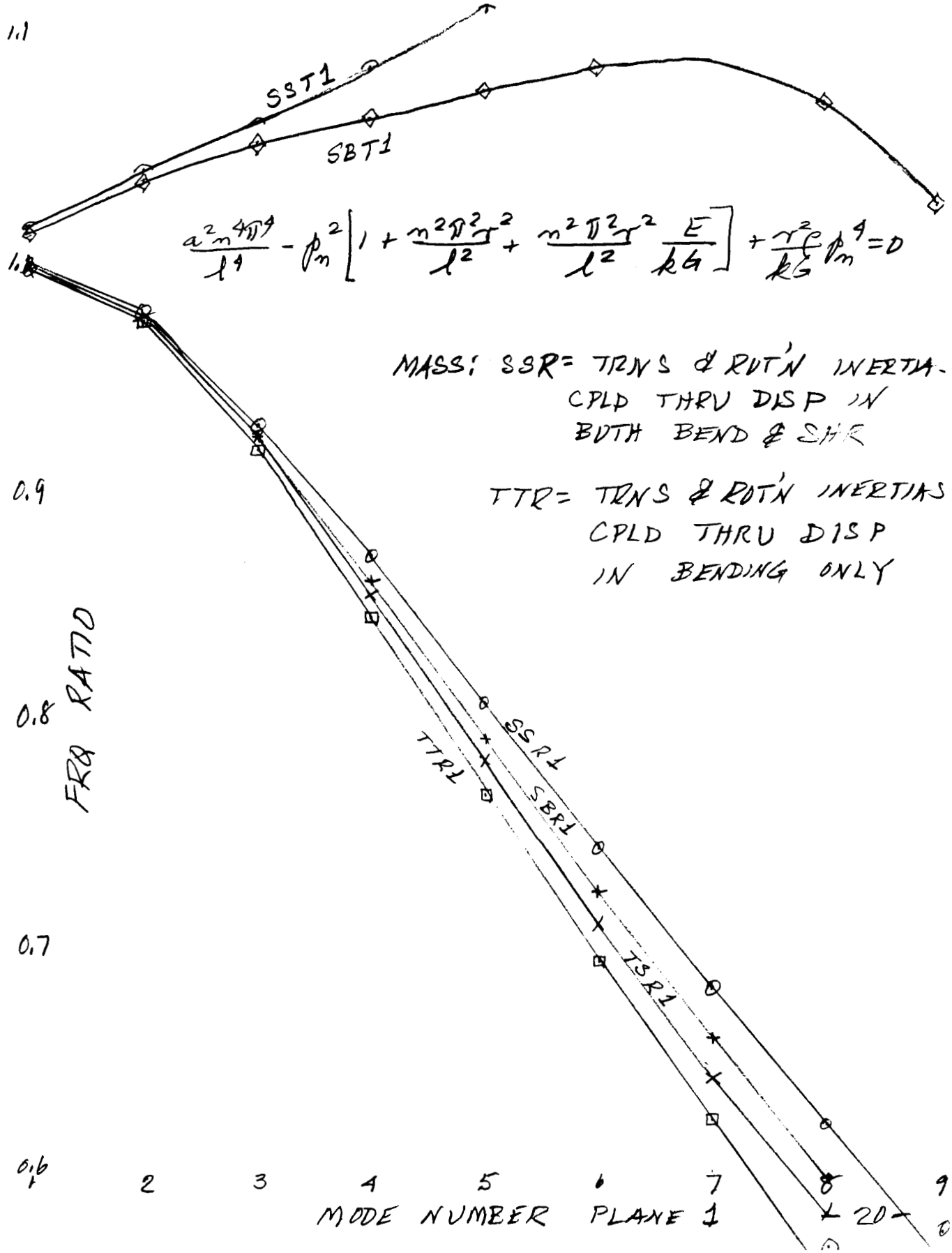


FIGURE 4

-19-


SST1
SBT1
$\frac{a^2 n^4 \pi^4}{l^4} - \beta_n^2 \left[1 + \frac{n^2 \pi^2 r^2}{l^2} + \frac{n^2 \pi^2 r^2}{l^2} \frac{E}{kG} \right] + \frac{r^2 \rho}{kG} \beta_n^4 = 0$
MASS: SSR= TRNS & ROT'N INERTIA-
CPLD THRU DISP IN
BOTH BEND & SHR
TTR= TRNS & ROT'N INERTIAS
CPLD THRU DISP
IN BENDING ONLY
FRQ RATIO
1.1
1.0
0.9
0.8
0.7
0.6
SSR1
SBR1
TSR1
TTR1
1
2
3
4
5
6
7
8
9
20
MODE NUMBER PLANE 1

STRUCTURAL OPTIMIZATION WITH ROCKWELL NASTRAN

Viney K. Gupta
NASTRAN Project Engineer

Rockwell International (Aerospace Operations)
Los Angeles, CA 90009, USA

SUMMARY

Computer-aided optimum design of large aerospace structures has traditionally employed coarse finite-element model (FEM) of a given configuration for preliminary design. Modernly, as presented herein, detailed models involving multi-level decomposition or substructuring may be optimally re-sized for minimum weight, subject to deflection, stress, buckling, and frequency constraints, in addition to FEM validation by improving test/analysis correlation. To reduce problem size during optimization, least-square orthogonal polynomials or bilinear shape functions may be employed to link design variables. In an effort to automate optimum design of composites, this capability has been developed for ROCKWELL NASTRAN, by integrating RPK Corp.'s CRAY version of NASA's COSMIC-released NASTRAN, the ADS nonlinear programming code, and NASA/AMES NASOPT interface program, on Rockwell's CRAY X-MP computer. Numerical examples tested demonstrate that proper use of optimization can be effective in a real-life environment, by promoting design economy within a timely schedule.

INTRODUCTION

With the primary objective of minimizing structural weight, cost (including producibility), or (1.0 - reliability), as a single-valued function, the design optimization module has been designed, developed, and implemented in ROCKWELL NASTRAN to generate a series of designs until convergence to the best design, in a minimum number of design iterations. The number of trial designs determined should be governed by whether or not the cost of new calculation would exceed savings resulting from the improved design. To do this manually would be difficult, time-consuming, and expensive. With computers becoming larger in storage, faster in turnaround, and cheaper

in price, it has become cost-effective to determine optimum solution quicker with better limits on design variables, geometric shapes can be refined, feasible mass and temperature-dependent modulus can be derived from experimental test data through close correlation with analytical FEM predictions. However, the ability to formulate problems is limited based on design experience, especially with composites. In a future implementation, the discrete optimization problem will have discrete data for design variables from spare part-size inventory, e.g., w.r.t. prescribed orientations for layered composite fibres, and the number of layers; parametric studies often yield the immediate answer at the present.

Rockwell NASTRAN finite-element structural analysis program, based on NASA's COSMIC-released NASTRAN code (1), has been augmented with ADS Constrained Optimization Code (2,3) to automate optimum structural design in a cost-effective manner. The program minimizes structural weight by reducing thickness, etc. for predefined shape/design configuration. The program is also useful to adjust the finite element model stiffness and mass for validation against test data, e.g. on frequencies and mode shapes of vibration, or if already validated, to reduce vibration by detuning based on selective structural modification (4-8). Other comparable codes are - ASTROS (9), STARSTRUC (10), NISA (11), ANSYS (12,13), MSC/NASTRAN (6), CSAR/OPTIM (14), and STARS (15). A typical problem in structural optimization consists of 200 design parameters, 2000 constraints, and bounds on design variables to yield a producible design among the multiple local optima discoverable by solving the mathematical programming problem using reasonable starting designs, which must be obtained first through either fully stressed design (1,13,14), stress ratio method (15), or optimality criteria formulation (10,16-22).

PROBLEM FORMULATION

As summarized in Table 1, the problem of structural optimization is formulated as a mathematical programming problem. The objective is to minimize weight, to validate FEM against test data with least structural change, or to reduce vibration under transient or harmonic excitation. For the test/analysis correlation problem, the objective represents a weighted sum of the squares (Euclidean norm) of the errors between test and analytical frequencies and/or eigenvectors. If the test data is incomplete or uncertain, other approaches (4-8,23) seem more practical. If the optimum structure is a minimum weight structure of specified natural frequencies, a strain energy density approach is often adopted. Elements possessing maximum

strain energy density are stiffened to seek the desired frequency separation between excitation frequency and the natural frequencies. To design composite structures (21,24,25) for minimum weight, detailed models involving substructures (26) may be optimally re-sized (laminae thickness, assuming discrete number of layers and fibre orientation angles), subject to deflection, stress, buckling, and frequency constraints, in addition to constraints on lamina failure indices and stresses. The design can involve thickness, area, moment of inertia, mass density, but excludes grid coordinates or shape variables, and flutter derivatives in the present code; nor does it optimize the dynamic response which may involve time-dependent constraints over certain period, or mean stress (integral-type): the equations of motion constitute an initial-value problem. We do not address the shape optimization problem (27-30) with a domain defined by a polynomial with unknown coefficients as design parameters, which NISA and ANSYS can probably do.

Problem Statement

Let (g(i), i=1,..,m) and (h(i), i=1, ..,p) be sets of functions defined on the n-dimensional Euclidean vector space Rn with values in the real space R. Let D = (Z: Z $\in$ Rn, g(i,Z) GE 0 for every i=1,..,m and h(i,Z)= 0 for every i=1,..,p). The mathematical programming problem is to determine Z* $\in$ D such that f(Z*) = glb(f(Z): Z $\in$ D), where f(Z*) is the value of the objective function f at the global point Z*.

Penalty Function Approach

Transformation methods transform the constraints and objective function to an unconstrained problem by means of an exterior or interior penalty function. We have modified our copy of the ADS fortran source code to enable optional use of the interior logarithmic penalty function, shown by Gupta (31) to be effective in solving the problem of constrained optimization problem as an unconstrained one:

$$P(Z,r) = f(Z) - r\sum \log(g(i)) + (1/r) \sum h(j)*h(j)$$

where r is a non-negative scalar initialized such that the inequality penalty term is an order of magnitude smaller than the objective function. The problem is most frequently non-convex in that the hessian of P(Z,r) w.r.t Z is semi-indefinite, yielding several local minima depending on the

starting design Z. A large condition number for the hessian matrix, i.e., the ratio of its largest to smallest eigenvalue can cause poor rate of convergence, particularly with the steepest descent method. The Davidon-Fletcher-Powell or BFGS variable-metric method in ADS code, on the other hand, is useful to generate an approximate hessian which is positive definite and preferrably well-conditioned for sake of convergence.

The interior-point penalty function formulation option in the ADS code is premised on maintaining feasibility of the design throughout the design iterations. Design parameters, e.g., thickness, area, moment of inertia, can be constrained to remain within user-specified limits. To estimate limits on state variables - maximum displacement, stress allowables, upper and lower bounds on frequency - is more difficult.

SENSITIVITY GRADIENTS

The analysis module solves equilibrium equations to predict response in terms of analysis variables - displacements and frequencies - which are implicit and nonlinear functions of design parameters. It provides values of the objective function and constraints and their gradients, separately; e.g., gradients disclose how sensitive stress is w.r.t. parameters. Analytical differentiation is laborious, error-prone, and difficult to implement. Though costly and dependent upon problem size and number of design variables, implicit differentiation (24,32) by perturbation of the equilibrium equations has been made possible in Rockwell's COSMIC-released NASTRAN; it reproduces the results of MSC/NASTRAN, following the same mathematical procedure (24,32). The design space method is not as efficient as the state space method, but chosen as in MSC/NASTRAN for ease of implementation. It provides the optimization code with first derivative or the Jacobian of the constraints and objective w.r.t. set of selected design variables. The scalar objective function represents the weight along the in-plane or normal (bending) direction, when seeking minimum-weight design.

Specifically, in implicit differentiation, assuming first-order perturbation, change in element stiffness matrix is computed, and then multiplied by the original static displacement solution to obtain pseudo loads. The displacement increment due to pseudo nodal loads is solved for using the original stiffness matrix, and is added to the original displacement solution to obtain the perturbated displacement solution for recovery of constraint and stress values.

OPTIMIZATION STRATEGY

Several Fortran codes - IDESIGN (33,34), CONLIN (24), NEWSUMT (30), and ADS (2) - are commercially available, for interface with your FEM code, or to solve mathematical programming problems with constraints. For unconstrained optimization, IMSL and NAG libraries provide several different gradient routines. The ADS code was selected for incorporation into ROCKWELL NASTRAN, in continuation of work with its predecessor CONMIN and its proclaimed success in ASTROS (9) with the modified method of feasible directions.

Optimization steps involved with the use of ROCKWELL NASTRAN and NASOPT/ADS codes that may be repeated to achieve an acceptable optimum design are as follows:

Step 1: Fully Stressed Design (FSD)

FSD module in NASTRAN was improvised to generate for ADS a good starting design to assure convergence to an acceptable minimum-weight design. Each element is re-sized in ratio of its strain-energy density, sqrt(E/M), where E is % of total strain energy, and M the % of total structure mass, to reach the prescribed displacement or stress limit, as if, decoupled from other elements.

The stress ratio method in FSD was greatly improved by making each element independent; its cross-sectional area and inertia were adjusted to remain mutually compatible and within prescribed lower and upper limits, based on linear stress ratio of its material stress allowable to that calculated for its current section modulus and area.

Step 2: FEM Analysis (Outer Loop)

Static analysis for minimum-weight design or an eigenvalue analysis for FEM validation is a pre-requisite for providing ADS optimization code with values of objective, constraints, gradients, and selected design parameters.

Step 3: Design Variable Linking

It is cost-effective to optimally re-size upto 100 independent design parameters using ADS code to achieve minimum-weight design or to validate FEM Model to closely approximate dominant

test frequencies and mode shapes. The extra variables can be made dependent through design variable linking, based on our new capability to automatically generate bilinear shape functions (35). The use of least-square orthogonal polynomials looks promising for the future (36).

Step 4: ADS Optimization (Inner Loop)

The independent design parameters are iteratively re-sized using initially-supplied gradients by the modified ADS code until convergence to a local minimum. The NASTRAN cards with the new design variable values are prepared to repeat as needed the expensive analysis of step 2, which the user can monitor and control, interactively.

EXAMPLE PROBLEMS

Table 2 lists some demonstration problems that have been successfully solved using Rockwell NASTRAN's newly implemented optimization capability based on the ADS code. Rockwell NASTRAN Group assists users with specific problem formulation and solution strategy selection.

NUMERICAL CONSIDERATIONS

A linear elastic structure is assumed here. If structure requires nonlinear analysis, the incremental design (37) sensitivity approach in ANSYS enables optimum design. More than one minimum solution may be found, because most structural problems are inherently not convex in nature in that they do not lend themselves to the globally convergent unique optimum solution from an arbitrary or random starting design, but instead rely on the proximity of the starting design. Numerical Considerations for the optimization technique require attention to generality, reliability of global convergence, efficient rate of convergence, variable step size to minimize the number of trials, selection procedure for potentially active constraints, ease of use, and designer-friendliness.

Convergence to Optima

Convergence to an optimum solution within ADS is indicated when relative change in design parameters is less than the user-specified tolerance (2%,say) in order to reduce the

objective further. Linear rate of convergence occurs with methods of Feasible direction, gradient projection, generalized reduced gradient and cost function bounding techniques. Potentially active constraints at current design iteration are included, the rest deleted from the next iteration to save cost. Feasible direction method zigzags with equality constraints. The variable-metric method with our implementation of the logarithmic penalty function tends to be more effective. However, for large structural problems, Vanderplaats (2) recommends Sequential linear, quadratic, or convex programming option for ISTRAT in Table 1 along with polynomial interpolation for 1-D search instead of using Penalty function and Golden section search, in an effort to reduce the number of NASTRAN FEM analyses. Generally, the implementation requires defining zero, heuristic rules to handle adverse situations based on user experience and safety measures, so as to seek convergence for ill-conditioned cases, since theory is valid in exact arithmetic not with limited arithmetic precision.

The quadratic programming problem with a quadratic cost function and linear constraints (especially with reciprocal variables: y = PL**3/3EI - nonlinear constraint becomes somewhat linear w/o need for a push-off factor in ADS) offers superlinear convergence with variable-metric methods for the constrained optimization problem, in which linear constraints represent Taylor series expansion of the given nonlinear constraints around current design state. Pshenichny's linearization (33) of constraints has been recommended. The superlinear rate occurs with the quadratic programming method.

Interactive Design Iterations

Interactive finite-element modelling facilitates verification with automated easy-to-use mesh-generation and refinement. Likewise, interactive design optimization allows design decision making and monitoring of algorithm interactively, but intuition or some experience is necessary. It can be slow for large problems, need supercomputers at back end, and can save valuable resources by proper monitoring of algorithms. One may introduce new design variables, objective and constraint functions, utilizing constraint violation histories and interactive graphics. Ability to choose algorithm, to restart from any design, to manually change design, or to fix design parameters for later release, are desirable features.

The new software trend will be dictated by problem size, lessons learnt or knowledge-base (38,39) in design optimization, data

base management (40) concepts, modular programming, efficiency and robustness, and parallel processing (41-43). Grooms, Merriman, and Hinz (44) are developing an expert system for training with Rockwell NASTRAN.

Quality of Optima

The post-optimization investigation may test how sensitive the optimum design is to constraints, how flat is the optimum w.r.t. design variables, i.e., in classifying sensitive and insensitive variables, relaxing stress on constraints to deal with discontinuous feasible regions, inspecting lagrange multipliers for constraints, and using average stresses or gauss-point stresses to handle stress discontinuities.

CONCLUDING REMARKS

Referring to Table 1, the paper is intended to help engineers recognize peculiar issues and essential concepts useful for design optimization, particularly with ROCKWELL NASTRAN.

ROCKWELL NASTRAN, based on NASA's COSMIC-released NASTRAN, has been enhanced with the following new capabilities:

1. Design Sensitivity Derivatives of weight objective and of constraints on deflection, stress, frequency, and eigenvectors, w.r.t. design parameters.

2. Addition of ADS mathematical programming code for design optimization problems such as:

   - Minimum-Weight Structural Design, subject to user-specified constraints,
   - Reduced Vibration Design,
   - Improving test-analysis correlation by least structural modification, or
   - FEM Model Validation.

3. Addition of Design Variable Linking capability based on bilinear shape functions.

Several example problems with complete JCL set-ups have been tested to support production use by Rockwell engineers.

REFERENCES

1. The NASTRAN Programmer's Manual (Level 17.5), National Aeronautics and Space Administration (NASA), Washington, D.C., NASA SP-223(05), Dec. 1978; The NASTRAN User's Manual, NASA SP-222(08), June 1986, COSMIC, University of Georgia, Athens, GA.

2. Vanderplaats, G. N., "ADS - A Fortran Program for Automated Design Synthesis, Ver. 1.10," May 1985, Engineering Design Optimization, Inc., Santa Barbara, CA.

3. Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design, McGraw-Hill, 1984.

4. Sciarra, J. J., "Vibration Reduction by Using Both the Finite Element Strain Energy Distribution and Mobility Techniques," Shock and Vibration Bulletin, Aug. 1974, pp. 193-199.

5. Gupta, V. K., and Marrujo, F. G., "Minimizing Unbalance Response of the CRBRP Sodium Pumps," Trans. 5th International Conference on Structural Mechanics in Reactor Technology (SMiRT), Paper F8/1, Aug. 1979; U.S. Dept. of Energy, Technical Review, Fall 1979, Report CRBRP-PMC 79-04.

6. Chargin, M, and Miura, H., "Dynamic Response Optimization Using MSC/NASTRAN," 1987 MSC World User's Conference, Mar. 11-12, 1987, Universal City, CA.

7. Hughes, P. C., "Space Structure Vibration Modes: How many exist? Which ones are important?," Proceedings of the Workshop on Applications of Distributed System Theory to the Control of Large Space Structures, NASA JPL Publication 83-46, pp. 31-48.

8. Walsh, J. L., "Optimization Procedure to Control the Coupling of Vibration Modes in Flexible Space Structures," Proceedings of the 28th AIAA/ASME/ASCE/AHS SDM Conf., Paper No. 87-0826, Monterey, CA, April 6-8, 1987.

9. Neill, D. J., Johnson, E. H., and Canfield, R., "ASTROS - A Multidisciplinary Automated Structural Design Tool," presented at the 28th AIAA/ASME/ASCE/AHS SDM Conf., Paper 87-0713, Monterey, CA, April 6-8, 1987.

10. Elsaie, A. M., Gatchel, S. G., Tabarrok, B., and Fenton, R. G., "STARSTRUC - A General Purpose Structural Optimization Program," University Computing Company, 1984.

11. Kothawala, K. S., "NISAOPT," Engineering Mechanics Research Corp. (EMRC), Troy, Michigan.

2. Vanderplaats, G. N., "Effective Use of Numerical Optimization in Structural Design," 1987 ANSYS Conf. Proceedings, Newport Beach, Calif., Mar. 31-April 2, 1987.

3. Swanson, J. A., and Marx, F. J., "Design Optimization Including Integrated Modelling Using the Finite Element Program ANSYS," presented at 1985 National OEM Design Show and Conf., Philadelphia, PA, Sept. 9-11, 1985.

4. Narayanaswami, R., and Cole, J. G., "CSA/NASTRAN," Computerized Structural Analysis & Research Corp., Northridge, CA.

5. Wellen, H., and Hertel, K., "Industrial Application of Structural Optimisation in Aircraft Construction: with STARS at MBB," PAFEC, Inc., Atlanta, GA, 1986.

16. Venkayya, V. B., and Tischler, V. A., "OPTSTAT: A Computer Program for the Optimal Design of Structures Subjected to Static Loads," WPAFB/FDL Technical Memorandum FBR-79-67, Wright-Patterson Air Force Base, Dayton, Ohio.

17. Oluyomi, M. A., and Tabarrok, "A Generalized Energy Approach to the Optimum Design of Plates and Skeletal Structures," Computers and Structures Jnl., Vol. 10, No. 1/2, 1979, pp.269-275.

18. Tabak, E. I., and Wright, P. M., "A Generalized Optimality Criteria Method for the Automated Design of Large Structures," Computers and Structures Jnl., Vol. 10, No. 1/2, 1979, pp. 341-363.

19. Fleury, C., and Braibant, V., "Structural Optimization: A New Dual Method using Mixed Variables," Intl. Journal of Numerical Methods in Engineering, Vol. 23, 1986, pp. 409-428.

20. Fleury, C., and Schmit, L., "Dual Methods and Approximation Concepts in Structural Synthesis," NASA Contractor Report, 3226, 1980.

21. Isakson, G., Pardo, H., Lerner, E., and Venkayya, V. B., "ASOP-3: A Program for the Optimum Design of Metallic and Composite Structures subjected to Strength and Deflection Constraints," 18th Structures, Structural Dynamics & Materials Conference, San Diego, Calif., Mar. 21-23, 1977, pp. 93-100.

22. Berke, L., and Khot, N. S., "Use of Optimality Criteria Methods for Large-Scale Systems," AGARD Lecture Series on Structural Optimization, Oct. 10-18, 1974, AFFDL-TM-74-70-FBR, April 1974.

23. Kammer, D. C., "An Optimum Approximation for Residual

Stiffness in Linear System Identification," Presented at the 28th AIAA/ASME/ASCE/AHS SDM Conf., Monterey, CA, April 6-8, 1987.

24. Nagendra, G. K., and Fleury, C., "Sensitivity Analysis and Optimization of Composite Structures using MSC/NASTRAN," 1986 MSC/NASTRAN User's Conf., Pasadena, CA, Mar. 13-14, 1986.

25. Nagendra, G. K., and Fleury, C., "Sensitivity Analysis and Optimization of Composite Structures using MSC/NASTRAN," NASA Symposium on Sensitivity Analysis in Engineering, NASA Conf. Proc. 2457, Langley Research Center, Hampton, VA, Sept. 25-26, 1986,

26. Sobieski, J., and Barthelemy, J.-F., "Improving Engineering System Design By Formal Decomposition, Sensitivity Analysis, and Optimization," NASA Tech. Memo. 86377, Feb. 1985.

27. Botkin, M. E., Yang, R. J., and Bennet, J. A., "Shape Optimization Three-dimensional Stamped and Solid Automotive Components," The Optimum Shape: Automated Structural Design, Plenum Press, N.Y., 1986, pp. 235-262.

28. Morris, A. J., Foundations of Structural Optimization: A Unified Approach, John Wiley, 1982.

29. Fleury, C., "Shape Optimal Design by the Convex Linearization Method," Intl. Symposium "The Optimum Shape: Automated Structural Design", General Motors, Warren, Michigan, Sept.30-Oct.1, 1985.

30. Schmit, L. A., "Structural Synthesis - Its Genesis and Development," AIAA J., Vol. 19, NO. 10, 1981, pp. 1249-1263.

31. Gupta, V. K., "Computer-aided Synthesis of Mechanisms Using Nonlinear Programming (SUMT)," Trans. ASME, Journal of Engineering for Industry, Feb. 1973, pp. 339-344.

32. Haug, E. J., and Arora, J. S., "Applied Optimal Design," John Wiley, 1979.

33. Arora, J. S., and Thanedar, P. B., "Computational Methods for Optimum Design of Large Complex Systems," Optimal Design Lab., Univ. of Iowa, 1986.

34. Thanedar, P. B., Arora, J. S., Tseng, C. H., Lim, O. K., and Park, G. J., "Performance of Some SQP Algorithms on Structural Design Problems," Optimal Design Lab., Univ. of Iowa, Nov. 1985.

35. Hughes, T. J. R., and Tezduyar, T. E., "Finite Element Based upon Mindlin Plate Theory With Particular

Reference to the Four-Node Bilinear Isoparametric Element," Trans. ASME, Journal of Applied Mechanics, Sept. 1981, pp. 587-596.

36. Yakowitz, S., "An Introduction to Numerical Computations," Macmillan Publishing Co., 1986.

37. Santos, J. T. L., and Choi, K. K., "Design Sensitivity Analysis of Nonlinear Structural Systems with an Established Finite Element Code," 1987 ANSYS Conf. Proceedings, Newport Beach, Calif., Mar. 31-Apr. 2, 1987.

38. Sobieszczanski-Sobieski, J., "Recent Experiences in Multidisciplinary Analysis and Optimization," NASA Conference Publication 2327, April 1984.

39. Rogers, Jr., J. L., and Barthelemy, M., "An Expert System for Choosing the Best Combination of Options in a General-Purpose Program for Automated Design Synthesis," Presented at 1985 International Computers in Engineering Conf. and Exhibition, Aug. 4-8, 1985, Boston, Mass.

40. Song, J. A., "Integrated Optimal Structural Design System Using a Relational Database Management System," Proceedings of the 28th AIAA/ASME/ASCE/AHS SDM Conf., Monterey, CA, April 6-8, 1987, Paper No. 87-0832, pp. 571-578.

41. Allik, H., Crowther, W., Goodhue, J., Moore, S., and Thomas, R., "Implementation of Finite Element Methods on the Butterfly (TM) Parallel Processor," Proceedings of the 1985 ASME International Computers in Engineering Conference and Exhibition, Aug. 4-8, 1985, Boston, Massachusetts.

42. The FLEX/32 Multicomputer System Overview, Flexible Computer Corp., Dallas, Texas.

43. Stanley, G. M., Felippa, C. A., Cabiness, H. D., Regelbrugge, M. E., and Weiler, F. C., "Preliminary Development of a Testbed for Computational Structural Mechanics, Part 1: The NICE/SPAR Prototype," Lockheed Palo Alto Report LMSC-D067201, June 1986, for NASA Langley Research Center, Hampton, Virginia.

44. Grooms, H. R., Merriman, W. J., and Hinz, P. J., "An Expert/Training System for Structural Analysis," ASME Pressure Vessel and Piping Conf., New Orleans, Lousiana, June 23-27, 1985.

****

****

TABLE 1 DESIGN OPTIMIZATION METHODOLOGY

****

FORMULATION (presently assumes linear elastic structure)

OBJECTIVE

Minimize Structural Weight (following Fully-Stressed Design)
Reduce Vibration (detune frequencies)

Validate FEM Model (w.r.t. disp/stress/freq)
Improve Test/Analysis Correlation (match freq/mode shapes)

Future: Optimize shapes (fillet, lugs,), controls problem, robotic (mechanisms), dynamic response, nonlinear static and dynamic analyses, and flutter optimization.

DESIGN PARAMETERS

Member Sizing - axial, membrane, bending, torsional
Cross-sectional Properties: A, I1, I2, J

Material Properties: moduli, mass density
Composite laminae thickness, orientation angles

Future: discrete variables, kinematic variables, control parameters, flutter parameters.

CONSTRAINTS (classify regions, elements, materials in groups)

Minimum and Maximum skin gauges, thicknesses, areas, etc.
Allowable deflections and stresses and failure indices

Frequency and buckling

Future: strains, number of plies, radii, discrete orientations, kinematic constraints, and flutter speed.

NASTRAN FEM ANALYSIS (& sensitivity gradients)

STATIC ANALYSIS (implicit differentiation - Haug/Arora technique)

EIGENSOLUTION (Mode Shape derivatives - Nelson's Method)

Future: enhance finite element using p-version technology, enhance flutter and robotic analysis capability, develop nonlinear design sensitivity capability, make gradient computation more cost-effective by pre-linking design parameters or optimality criteria.

****

TABLE 1 DESIGN OPTIMIZATION METHODOLOGY (Continued)

ADS OPTIMIZATION (solve mathematical programming problem)

LINK DESIGN VARIABLES (bilinear shape functions)

Future: Least-square orthogonal polynomials

RETAIN ACTIVE CONSTRAINTS (Taylor series local linearization)

Future: improve for nonlinear constraints/reciprocal variables

SELECT ADS OPTIONS (typical)

STRATEGY (ISTRAT)	OPTIMIZER (IOPT)	IONED (1-D Search)
SUMT, Linear Penalty	DFP Var. Metric	Golden Section
SUMT, Logarithmic	BFGS Var. Metric	Poly. Int. bounds)
Sequential Linear Prog.	DFP Var. Metric	Poly. Int. bounds)
Sequential Quadr. Prog.	DFP Var. Metric	Poly. Int. bounds)
Sequential Convex Prog.	DFP Var. Metric	Poly. Int. bounds)
None	Mod. Feas. Dir.	Poly. Int. bounds)

SOLVE FOR CONVERGED DESIGN PARAMETERS

Future: enhance global convergence and reliability

RE-FORMULATE OR REPEAT NASTRAN AND ADS ANALYSES

Future: interactive design optimization with integrated system including DBMS, Expert Systems, automated graphics, and parallel processing.

****

TABLE 2 OPTIMIZATION DEMONSTRATION PROBLEMS

****

STATIC WEIGHT OPTIMIZATION

- GPWG CALCULATES WEIGHT AND ITS DERIVATIVES FOR X1 (MEMBRANE) OR X3(BENDING) W.R.T. DESIGN VARIABLES
- AIRCRAFT VERTICAL TAIL PROBLEM
- AIRCRAFT WING PROBLEM

FULLY-STRESSED DESIGN COMPOSITE WING PROBLEM WITH CQUAD4/CTRIA3

DESIGN VARIABLE LINKING VERIFICATION PROBLEM

- LINEAR OR BILINEAR SHAPE FUINCTIONS

TEST/ANALYSIS CORRELATION PROBLEM (frequencies & mode shapes)

- FREQUENCY CONSTRAINTS

****

Effect of Element Size on the Solution Accuracies of Finite-Element Heat Transfer and Thermal Stress Analyses of Space Shuttle Orbiter

William L. Ko and Timothy Olona
Ames Research Center, Dryden Flight Research Facility, Edwards, California

SUMMARY

The effect of element size on the solution accuracies of finite-element heat transfer and thermal stress analyses of space shuttle orbiter was investigated. Several structural performance and resizing (SPAR) thermal models and NASA structural analysis (NASTRAN) structural models were set up for the orbiter wing midspan bay 3. The thermal model was found to be the one that determines the limit of finite-element fineness because of the limitation of computational core space required for the radiation view factor calculations. The thermal stresses were found to be extremely sensitive to a slight variation of structural temperature distributions. The minimum degree of element fineness required for the thermal model to yield reasonably accurate solutions was established. The radiation view factor computation time was found to be insignificant compared with the total computer time required for the SPAR transient heat transfer analysis.

NOMENCLATURE

C	capacitance matrix
CQUAD2	quadrilateral membrane and bending element
CROD	two-node tension-compression-torsion element
C41	four-node forced convection element
E23	bar element for axial stiffness only
E25	zero length element used to elastically connect geometrically coincident joints
E31	triangular membrane element
E41	quadrilateral membrane element
E44	quadrilateral shear panel element
F_{ij}	view factor from element i to element j
FRSI	felt reusable surface insulation
H	convection load vector
HRSI	high-temperature reusable surface insulation
JLOC	joint location
K_{h}	convection matrix
K_{k}	conduction matrix
K_{r}	radiation matrix
K21	two-node line conduction element
K31	three-node area conduction element
K41	four-node area conduction element
K81	eight-node volume conduction element
NASTRAN	NASA structural analysis
Q	source load vector
R	radiation load vector
R31	three-node area radiation element
R41	four-node area radiation element
SIP	strain isolation pad
SPAR	structural performance and resizing
STS	space transportation system
T	absolute temperature, °R
TPS	thermal protection system
t	time, sec

x, y, z	rectangular Cartesian coordinates
X_0	station on x axis, in
Y_0	station on y axis, in
σ_x	normal stress in x direction (chordwise stress), ksi
σ_y	normal stress in y direction (spanwise stress), ksi
τ_{xy}, τ_{yz}	shear stresses, ksi

INTRODUCTION

In finite-element heat transfer analysis or finite-element stress analysis, it is well known that reduction of element sizes (or increase in element number) will improve the solution accuracy. For simple structures, the element sizes may be reduced sufficiently to obtain highly accurate solutions. However, for large complex structures, such as the space shuttle orbiter, the use of excessively fine elements in the finite-element models may result in unmanageable computations that exceed the memory capability of existing computers. This computational limitation is frequently encountered during radiation view factor computations in the three-dimensional finite-element heat transfer analysis of complex structures. Because of computational limitations in the past heat transfer analysis of the space shuttle orbiter, only small local regions of the orbiter structure were modeled. Several regions of the space shuttle were modeled by Ko, Quinn, and Gong. For the past several years, these finite-element models were used to calculate orbiter structural temperatures, which were correlated with the actual flight data during the initial orbit tests of the space shuttle Columbia (refs. 1 to 7). Recently, Gong, Ko, and Quinn (ref. 4) conducted a finite-element heat transfer analysis of the orbiter whole wing (fig. 1) using a thermal model with relatively coarse elements (fig. 2). A similar whole wing finite-element structural model was used by Ko and Fields (ref. 8) in the thermal stress analysis of the orbiter whole wing. Both the thermal model (fig. 2) and the corresponding structural model (fig. 3) set up for the orbiter whole wing were too coarse to give sufficiently accurate structural temperature and thermal stress distributions. Before modifying the existing wing models by increasing the number of joint locations to improve the solutions, it is necessary to determine the minimum number of joint locations required for the modified wing thermal model (the corresponding wing structural model requires far fewer joint locations) to give reasonably accurate structural temperature distributions without causing the radiation view factor computations to become unmanageable. This report describes (1) heat transfer and thermal stress analyses of a single bay at the orbiter wing midspan using several different thermal and structural models having different numbers of joint locations (or different element sizes), (2) the effect of element sizes on the accuracies of solutions, and (3) the minimum number of joint locations required for the single-bay model to give reasonably accurate solutions. The results of this report will form the basic criteria in remodeling the whole orbiter wing or modeling other types of hypersonic aircraft wings (hot structures).

WHOLE WING THERMAL AND STRUCTURAL MODELS

In finite-element thermal stress analysis of the space shuttle orbiter, the temperature input to the structural model for the calculation of thermal stresses is usually obtained from the results of finite-element (or finite-difference) heat transfer analysis using the corresponding thermal model. Since the thermal protection system (TPS) is not a major load-carrying structure, it is neglected in the structural model. Thus, the structural model has far fewer joint locations (JLOCs) than the corresponding thermal model. For the wing models, the thermal model contains 2289 JLOCs, while the structural model has only 232 JLOCs (see table 1). Even though the thermal model has only one degree of freedom (temperature), because of

the radiation view factor computations and the transient nature of heat transfer, the computer core space required by the thermal model is always many times more than that required for the structural model, which has six degrees of freedom. Thus, the thermal model is the one that limits how fine the element size can be reduced for improving the solutions.

ONE-CELL THERMAL MODELS

To study the improvement of structural temperature distributions by reducing the element sizes, and also to study the associated effort involved in the computations of radiation view factors, five structural performance and resizing (SPAR, ref. 9) finite-element thermal models (with different degrees of element fineness) were set up for the orbiter wing midspan bay 3 bounded by Y_0-226 and Y_0-254 (see fig. 1). The five SPAR thermal models A, B, C, D, and E are shown in figure 4. The thermal model A is set up to match the coarseness of the existing whole wing thermal model. The four-node area conduction (K41) elements were used to model the wing skins, spar webs, rib cap shear webs, room temperature vulcanized (RTV) rubber layers lying on both sides of the strain isolation pad (SIP), and TPS surface coatings. The aerodynamic surfaces for providing source heat generation were modeled with one layer of K41 elements of unit thickness. The spar caps, rib caps, and rib trusses were modeled with two-node line conduction (K21) elements. The TPS was modeled in 10 layers on the lower surface and 3 layers on the upper surface using eight-node volume conduction (K81) elements. The SIP layer was modeled with only one layer of K81 elements. The external and internal radiations were modeled by attaching a layer of four-node area radiation (R41) elements to the active radiation surfaces. The radiation into space was modeled with one R41 element of unit area. No radiation elements were attached to the surfaces of spar caps, rib caps, rib cap shear webs, and rib trusses because of small exposed areas. A layer of four-node forced convection (C41) elements were attached to the internal surfaces of the bay to model the internal convection of air resulting from the entrance of external cool air into the interior of the orbiter wing at 1400 sec after reentry (or at 100,000 ft altitude). The front and rear ends of the thermal models were insulated. Table 2 summarizes the sizes (joint location number, number of different types of elements) of the five SPAR thermal models A, B, C, D, and E.

Heat Input

The external heat inputs to the SPAR thermal models are shown in figure 5. These aerodynamic heating curves are associated with STS-5 flight trajectories and are taken from reference 4, which describes in detail the method of calculations of aerodynamic heating.

View Factors

The view factors used in the radiation to space were calculated by hand. However, for the internal radiation exchanges, the view factors were calculated by using a VIEW computer program, which is incorporated into the SPAR thermal analysis computer program (ref. 9).

For both the external and the internal thermal radiation exchanges, all the view factors were calculated from the equation (ref. 9)

$$A_i F_{ij} = A_j F_{ji} \tag{1}$$

where A_i is the surface area of radiation exchange element i and F_{ij} is the view factor, defined as the fraction of radiant heat leaving element i incident on element j. In the calculation of view factors for the external radiation exchanges (considering that element i represents the space element and element j any radiation exchange element on the wing surface), F_{ji} was taken to be unity; therefore, $F_{ij} = A_j/A_i$ according to equation (1).

Values of emissivity and reflectivity used to compute radiant heat fluxes are given in table 3. The initial temperature distribution used in the analysis was obtained from the actual flight data. In thermal modeling, the majority of the time was consumed in the computations of view factors.

Internal Forced Convection

After opening the landing gear door and the vents at the wing roots, external air enters the shuttle wing and induces convective heat transfer. The heat transfer coefficients used for C41 elements were calculated using the effective air flow velocities inside the wing, listed in table 4 (ref. 6).

Transient Thermal Solutions

The SPAR thermal analysis finite-element computer program was used in the calculation of temperature time histories at all joint locations of the thermal models. The SPAR program used the following approach to obtain transient thermal solutions.

The transient heat transfer matrix equation

$$(K_{\mathrm{k}} + K_{\mathrm{r}} + K_{\mathrm{h}})T + C\dot{T} = Q + R + H \tag{2}$$

where

K_{k}	is the conduction matrix,
K_{r}	the radiation matrix,
K_{h}	the convection matrix,
T	the absolute temperature,
C	the capacitance matrix,
Q	the source load vector,
R	the radiation load vector,
H	the convection load vector, and
$\dot{[\,]}$	denotes time derivative,

was integrated by assuming that the temperature vector T_{i+1} at time step t_{i+1} can be expressed as

$$T_{i+1} = T_i + \dot{T}_i\,\Delta t + \frac{1}{2!}\ddot{T}_i\,\Delta t^2 + \frac{1}{3!}\dddot{T}_i\,\Delta t^3 + \cdots \tag{3}$$

where T_i is the temperature vector at time step t_i and Δt is the time increment. The vector $\dot{T}_i$ is determined directly from equation (2) as

$$\dot{T}_i = -C^{-1}(K_{\mathrm{k}} + K_{\mathrm{r}} + K_{\mathrm{h}})T_i + C^{-1}(Q + R + H) \tag{4}$$

Higher order derivatives are obtained by differentiating equation (2) according to the assumptions that (1) material properties are constant over Δt, (2) Q and H vary linearly with time, and (3) R is constant over Δt:

$$\ddot{T}_i = -C^{-1}(K_{\mathrm{k}} + 4K_{\mathrm{r}} + K_{\mathrm{h}})\dot{T}_i + C^{-1}(\dot{Q} + \dot{H}) \tag{5}$$

$$\dddot{T}_i = -C^{-1}(K_{\mathrm{k}} + 4K_{\mathrm{r}} + K_{\mathrm{h}})\ddot{T}_i - 4C^{-1}\dot{K}_{\mathrm{r}}\dot{T}_i \tag{6}$$

In the present computations, the Taylor series expansion (eq. (3)) was cut off after the third term. The pressure dependency of the TPS and SIP thermal properties was converted into time dependency based on the trajectory of the STS-5 flight.

Time-dependent properties were averaged over time intervals (RESET TIME), which were taken to be 25 sec. Temperature-dependent properties were evaluated at the temperatures computed at the beginning of each time interval. The values Q, $\dot{Q}$, and R were computed every 2 sec.

ONE-CELL STRUCTURAL MODELS

For the thermal stress analysis, the NASA structural analysis (NASTRAN, ref. 10) computer program was used because it can handle temperature-dependent material properties. The SPAR structural computer program lacks this capability. The five NASTRAN structural models (not shown) corresponding to the five SPAR thermal models A, B, C, D, and E (fig. 4) are essentially the same except that the TPS layers are removed in the NASTRAN structural models. Thus, each set of thermal and corresponding structural models have identical joint locations so that the temperature distribution obtained from the thermal model can be input directly to the corresponding structural model for the calculations of thermal stresses. The wing skins, spar webs, and rib cap shear webs were modeled with quadrilateral membrane and bending (CQUAD2) elements. The spar caps, rib caps, and rib trusses were represented with two-node tension-compression-torsion (CROD) elements. To approximate the deformation field of the midspan bay 3 when it is not detached from the whole wing, the following boundary conditions were imposed on the NASTRAN structural models.

1. Y_0-226 plane fixed—The grid points lying in the Y_0-226 plane have no displacements in the y direction but are free to move in the x and z directions. The rotations with respect to the x, y, and z axes are constrained.

2. Y_0-254 plane free—The grid points lying in the Y_0-254 plane are free to move in the x, y, and z directions. The rotations with respect to the x, y, and z axes are constrained.

The thermal loadings to the NASTRAN structural models were generated by using the structural temperature distributions calculated from the corresponding SPAR thermal models. Table 5 summarizes the sizes of the five NASTRAN structural models. Because the TPS is removed, the structural models have far fewer joint locations as compared with corresponding SPAR thermal models (see table 2).

RESULTS

Structural Temperatures

Figure 6 shows the time histories of the midbay TPS surface temperatures calculated by using different SPAR thermal models. The five temperature curves respectively associated with the thermal models A, B, C, D, and E are so close as to be pictorially undiscernable. This implies that the element sizes in the substructure have negligible effect on the TPS surface temperatures. The STS-5 flight data are also shown in figure 6 (solid circles) for comparison. Figure 7 shows the time histories of the structural temperatures in the midbay regions of the lower and upper wing skins calculated from different thermal models. The thermal models B, C, D, and E yielded almost identical skin temperatures in the midbay regions. However, the thermal model A gave slightly lower wing skin temperatures because of coarseness of the model. The STS-5 flight data are also shown in figure 7 (solid circles) for comparison. Figure 8 shows the three-dimensional distributions of the wing skin temperatures, at t = 1700 sec from reentry, over whole surfaces of the lower and upper wing skins, calculated from different thermal models. The roof-shaped wing skin temperature distributions given by thermal model A (fig. 8(a)) is inadequate to represent actual distributions of the wing skin temperatures. The dome-shaped wing skin temperature profiles calculated from the thermal models B, C, D, and E (fig. 8(b) to (e)) are caused by the existence of the spars and ribs, which function as heat sinks. The dome-shaped wing skin temperature profiles imply the degree of thermal stress buildup in the wing skins, as will be discussed in the following section.

Figure 9 shows the calculated structural temperature distributions in the plane Y_0-240 of bay 3 at t = 1700 sec from reentry. The thermal model A definitely yielded inaccurate solutions. The structural temperature distributions given by thermal models B, C, D, and E are quite close. Especially, the thermal

models D and E yielded very close structural temperature distributions. As shown in the following section, a slight difference in the structural temperature distributions obtained from different thermal models could cause a "marked" difference in the induced thermal stress distributions. The structural temperature gradients are steepest near the lower spar caps because the spar webs function as heat sinks. Figure 10 shows the spanwise distributions of the wing skin temperatures at cross section $X_0 1270$ based on different thermal models. The thermal models B, C, and D yield almost identical structural temperature distributions because they have the same number of elements in the spanwise direction. The shapes of the skin temperature distributions given by model E approach circular arcs. The solutions given by the thermal model A are rather poor because of an insufficient number of elements. When the number of the finite elements is increased sufficiently, the ultimate structural temperature distributions in the midspan bay 3 look like the curves shown in figures 11 and 12. The curves in the figures were constructed by fitting the data points obtained from SPAR thermal model E with smooth continuous curves.

Thermal Stresses

Figures 13 to 15 respectively show the distributions of the chordwise stresses σ_x, spanwise stresses σ_y, and shear stresses τ_{xy} calculated using different NASTRAN structural models. Clearly the structural model A gave inaccurate stress predictions. For the wing lower skin, the models C and D give σ_y distribution with stress-release zone at the mid bay region (fig. 14(c) and (d)). The σ_y distribution given by model E (fig. 14(e)) exhibits two zones: (1) stress-release zone between $y = -240$ and $y = -254$ and (2) stress-increase zone between $y = -226$ and $y = -240$. Figure 16 shows distributions of the spanwise stress σ_y calculated by using different NASTRAN structural models. Notice that the thermal stresses are very sensitive to the finite-element sizes (or structural temperature distributions). The coarser models A and B yielded peak compression in the midbay regions of both lower and upper skins. However, as the number of elements increased (models C, D, and E), the shallow U-shaped distributions of σ_y in the lower skin shifted to shallow W-shaped distributions, and the peak compression regions moved near the spar webs. The slight stress release in the midbay region of the lower skin, based on the structural models C, D, and E, is due to the bulging of the wing skin (described later in this section). For the upper skin, the zone of slight stress release showed up only for the stress distributions calculated from models D and E. These stress releases in the midbay regions of the wing skins were never observed in the earlier thermal stress analysis, which ignored the three-dimensional deformations of the orbiter skins (that is, skin-bulging effect). Figure 17 shows the distributions of chordwise stresses σ_x calculated from the five structural models. Again, the solution given by the model A is quite poor. The distributions of σ_x given by the structural models B, C, and D (all of which have four elements in the spanwise direction) are quite close. The structural model E, which has eight elements in the spanwise direction, gave a magnitude of peak compressional stress about 1.2 ksi above those predicted from the structural models B, C, and D. The marked difference in the σ_x distribution given by model E and those given by models B, C, and D is due to the existence of a stress-increase zone, which appeared only in model E. Unlike the distribution of σ_y (fig. 16), the distributions of σ_x calculated from all structural models did not exhibit stress release effects in the midbay regions of the wing skins. The magnitude of thermal stress σ_x (either in tension or compression) is higher than that of thermal stress σ_y shown in figure 16. Thus, σ_x is more critical than σ_y because the buckling strength of the wing skin in the x direction (normal to the hat stringers) is lower than that in the y direction (parallel to the hat stringers). The orbiter wing skin buckling stresses are in the neighborhood of $\sigma_x = -12$ ksi (normal to hat stringers) and $\sigma_y = -25$ ksi (parallel to hat stringers).

Figure 18 shows the distributions of shear stresses τ_{xy} and τ_{yz} in the cross section Y_0-252 (plane of highest shear) predicted from different NASTRAN structural models. The high shear-stress regions are near the lower spar caps.

When the number of finite elements is increased sufficiently, the ultimate distributions of the thermal stresses in the midspan bay 3 will look like the curves shown in figures 19 to 21. Those curves in the figures were constructed by fitting the data points obtained from NASTRAN structural models E with smooth continuous curves. Figure 22 shows the deformed shape of the orbiter wing midspan bay 3 due to STS-5 thermal loading. The front half of the wing lower skin bulged inwardly, but the rear half bulged outwardly; almost the entire wing upper skin bulged outwardly with more severe deformations in the front half region.

Computation Time

Table 6 summarizes the number of internal radiation view factors F_{ij} needed for different SPAR thermal models, the total computation time used in the transient heat transfer analyses associated with each thermal model, and the radiation view factor computation time. The data shown in table 6 are plotted in figure 23. Both the SPAR computation time and the number of internal radiation view factors appear to increase almost exponentially with the increase in the number of JLOCs. However, the time required for the radiation view factor computations turned out to be insignificant compared with the total SPAR computation time. The curves in figure 23 show how fast the computational "barrier" will be reached by accelerating the increase in the number of JLOCs.

CONCLUSIONS

Finite-element heat transfer and thermal stress analyses were performed on the space shuttle wing midspan bay 3 using several finite-element models of different degrees of element fineness. The effect of element sizes on the solution accuracy was investigated in great detail. The results of the analyses are summarized as follows:

1. The finite-element model A (thermal or structural), which has the same coarseness as the earlier whole wing model, is too coarse to yield satisfactory solutions.

2. The structural temperature distribution over the wing skin (lower or upper) surface of one bay was "dome" shaped and induced more severe thermal stresses in the chordwise direction than in the spanwise direction. The induced thermal stresses were very sensitive to slight variation of structural temperature distributions.

3. The structural models with finer elements yielded spanwise stress distributions exhibiting a stress release zone (due to skin bulging) at the midbay region of the wing skin (lower or upper), and the peak wing skin compression occurred near the spar caps. However, the coarser models gave the peak skin compression in the midbay region.

4. The front half of the wing lower skin bulged inwardly, but the rear half bulged outwardly. Almost the entire wing upper skin bulged outwardly with more severe deformations in the front half region.

5. For obtaining satisfactory thermal stress distributions, each wing skin (lower or upper) of one bay must be modeled with at least 8 elements in the spanwise direction (model E) and 10 elements in the chordwise direction (model D); each spar web must be modeled with at least 5 elements in the vertical directions (model D).

6. Both the computation time required for the SPAR transient heat transfer analysis and the number of view factors needed for internal radiation computations appeared to increase almost exponentially with the increase of the number of joint locations.

7. Even with the huge number of radiation view factor computations, the radiation view factor computation time was found to be insignificant compared with the total computer time required for the SPAR transient heat transfer analysis.

REFERENCES

1. Ko, William L.; Quinn, Robert D.; Gong, Leslie; Schuster, Lawrence S.; and Gonzales, David: Preflight Reentry Heat Transfer Analysis of Space Shuttle. AIAA-81-2382, Nov. 1981.

2. Ko, William L.; Quinn, Robert D.; Gong, Leslie; Schuster, Lawrence S.; and Gonzales, David: Reentry Heat Transfer Analysis of the Space Shuttle Orbiter. NASA CP-2216, 1982, pp. 295–325.

3. Gong, Leslie; Quinn, Robert D.; and Ko, William L.: Reentry Heating Analysis of Space Shuttle With Comparison of Flight Data. NASA CP-2216, 1982, pp. 271-294.

4. Gong, Leslie; Ko, William L.; and Quinn, Robert D.: Thermal Response of Space Shuttle Wing During Reentry Heating. AIAA-84-1761, June 1984. (Also published as NASA TM-85907, 1984.)

5. Ko, William L.; Quinn, Robert D.; and Gong, Leslie: Finite-Element Reentry Heat Transfer Analysis of Space Shuttle Orbiter. NASA TP-2657, 1986.

6. Ko, William L.; Quinn, Robert D.; and Gong, Leslie: Effect of Forced and Free Convections on Structural Temperatures of Space Shuttle Orbiter During Reentry Flight. AIAA-87-1600, June 1987.

7. Gong, Leslie; Ko, William L.; and Quinn, Robert D.: Comparison of Flight-Measured and Calculated Temperatures on Space Shuttle Orbiter. NASA TM-88278, 1987.

8. Ko, William L.; and Fields, Robert A.: Thermal Stress Analysis of Space Shuttle Orbiter Subjected to Reentry Aerodynamic Heating.

9. Marlowe, M.B.; Moore, R.A.; and Whetstone, W.D.: SPAR Thermal Analysis Processors Reference Manual, System Level 16, Volume 1: Program Execution. NASA CR-159162, 1979.

10. The NASTRAN User's Manual, Level 17.5. NASA SP-222(05), 1978.

TABLE 1. COMPARISON OF FINITE-ELEMENT THERMAL AND STRUCTURAL MODELS FOR SPACE SHUTTLE ORBITER WING

Thermal model		Structural model	
Feature	Number	Feature	Number
JLOCs	2289	JLOCs	232
K21 elements	1696	E23 elements	498
K31 elements	84	E25 elements	10
K41 elements	485	E31 elements	19
R31 elements	84	E41 elements	181
R41 elements	568	E44 elements	67

TABLE 2. SIZES OF SPAR THERMAL MODELS

SPAR thermal model	JLOCs	Element				
		K21	K41	K81	R41	C41
A	112	34	28	28	15	10
B	436	54	168	224	89	56
C	636	82	232	336	137	88
D	972	98	360	560	201	120
E	2076	146	848	1344	513	320

TABLE 3. EMISSIVITY AND REFLECTIVITY VALUES USED TO COMPUTE RADIANT HEAT FLUXES

Surface	Emissivity	Reflectivity
Windward	0.85	0.15
Leeward	0.80	0.20
Internal structure	0.667	0.333
Space	1.0	0

TABLE 4. EFFECTIVE AIR FLOW VELOCITIES AND ASSOCIATED HEAT TRANSFER COEFFICIENTS FOR INTERNAL FORCED CONVECTION

Time (sec)	Effective air flow velocity (ft/sec)	Heat transfer coefficient (Btu/sec-in^2-°F)
1750	25	3.30×10^{-6}
1850	25	4.00×10^{-6}
2000	15	2.73×10^{-6}
3000	0	0.35×10^{-6}[a]

[a]Heat transfer coefficient for natural convection.

TABLE 5. SIZES OF NASTRAN STRUCTURAL MODELS

NASTRAN structural model	Grid	CQUAD2	CROD
A	24	18	54
B	82	72	54
C	140	112	74
D	196	160	90
E	429	368	132

TABLE 6. NUMBERS OF JOINT LOCATIONS AND INTERNAL RADIATION VIEW FACTORS AND THERMAL ANALYSIS COMPUTATION TIME ASSOCIATED WITH DIFFERENT SPAR THERMAL MODELS

SPAR thermal model	JLOCs	Number of internal radiation F_{ij}	SPAR computation time (min (hr))	F_{ij} computation time (min (hr))	Percent F_{ij} computation time
A	112	78	15 (0.25)	1.83 (0.031)	12.20
B	436	2,816	75 (1.25)	2.60 (0.043)	3.47
C	636	6,894	210 (3.5)	3.60 (0.060)	1.71
D	972	13,500	540 (9.0)	5.15 (0.086)	0.95
E	2076	93,869	1890 (31.5)	23.02 (0.384)	1.22

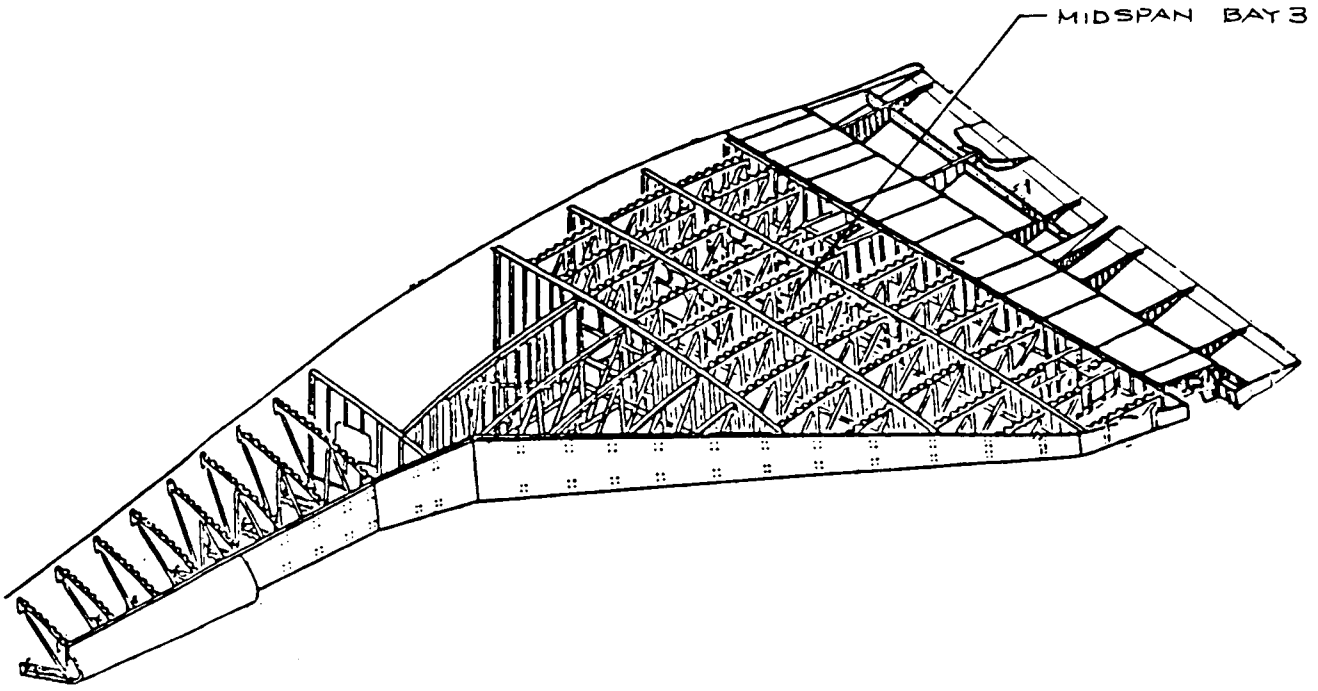


Figure 1. Space shuttle wing.

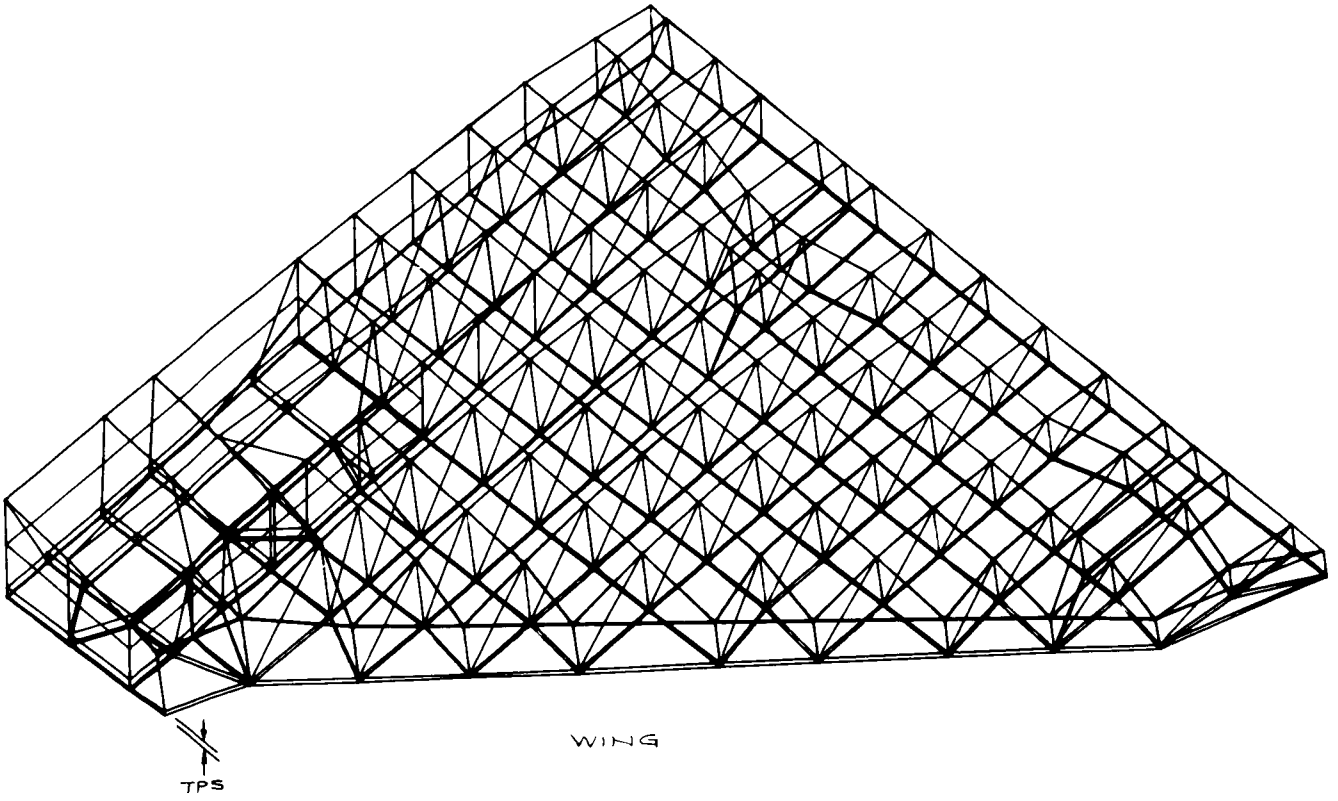


Figure 2. Space shuttle wing SPAR thermal model.

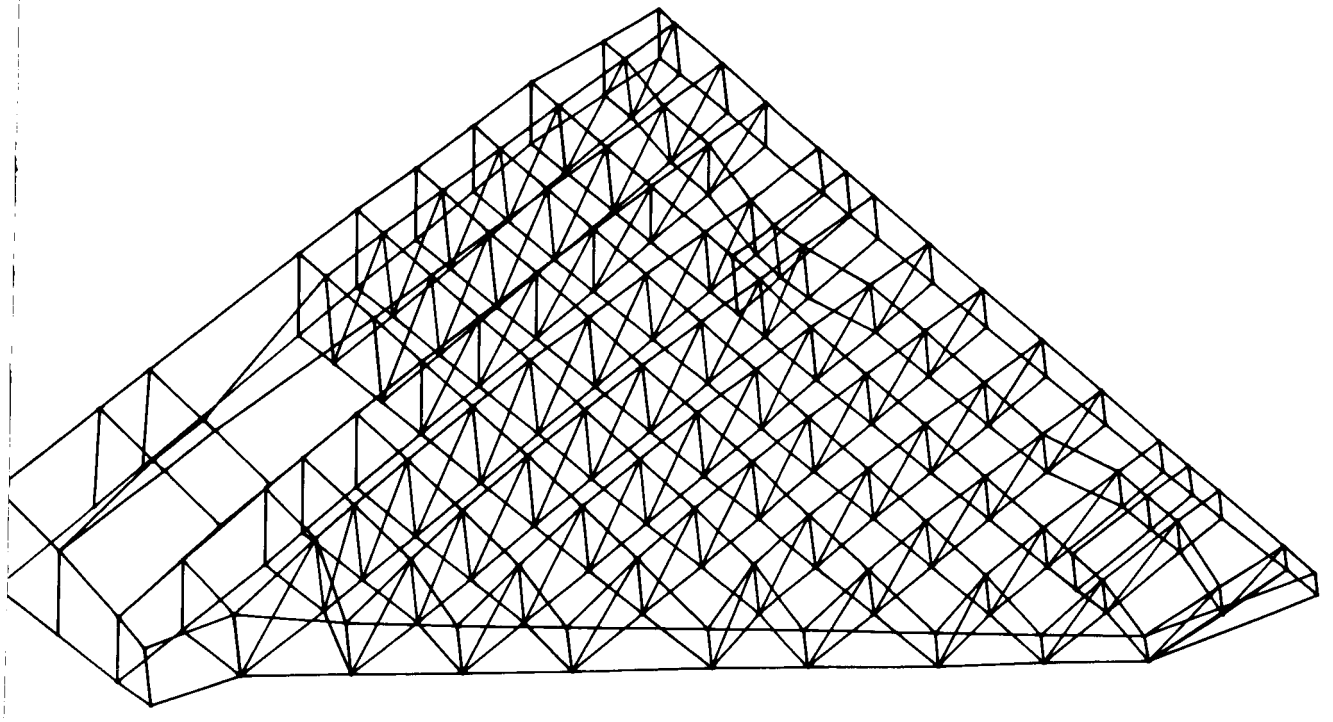
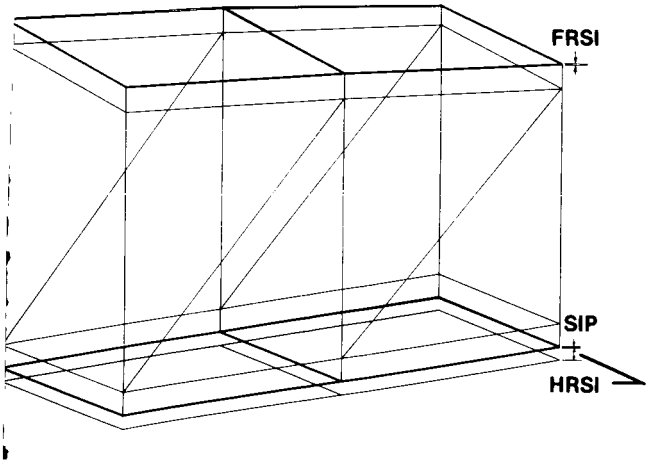
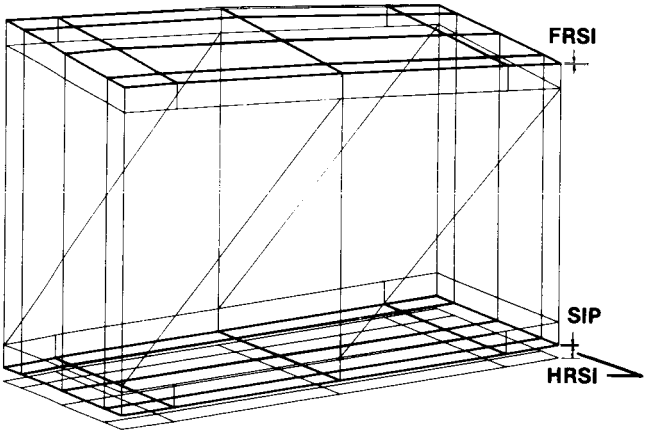


Figure 3. Space shuttle orbiter wing SPAR finite-element structural model. TPS, wheel well door, and landing gear excluded.

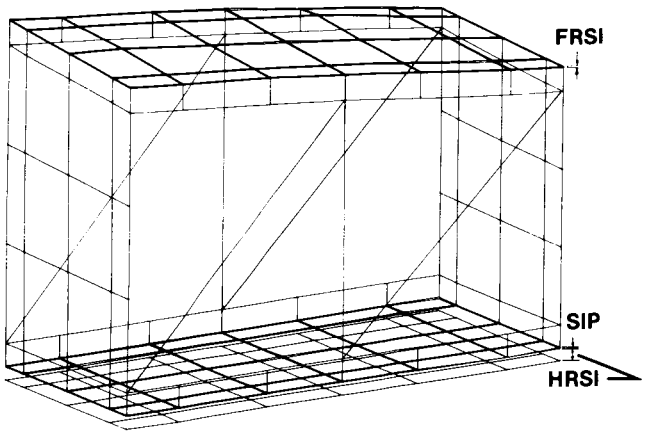


(a) Model A.

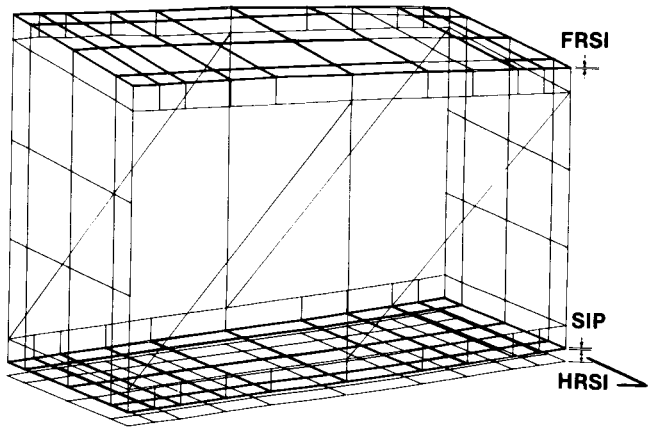


(b) Model B.

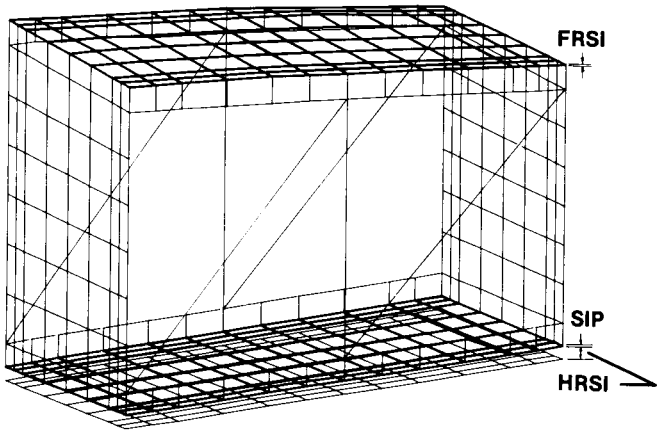
Figure 4. SPAR thermal models for bay 3 of orbiter wing bounded by Y_0-226 and Y_0-254. K81 elements for TPS and SIP not shown. TPS and SIP removed to convert to NASTRAN structural models.



(c) Model C.



(d) Model D.



(e) Model E.

Figure 4. Concluded.

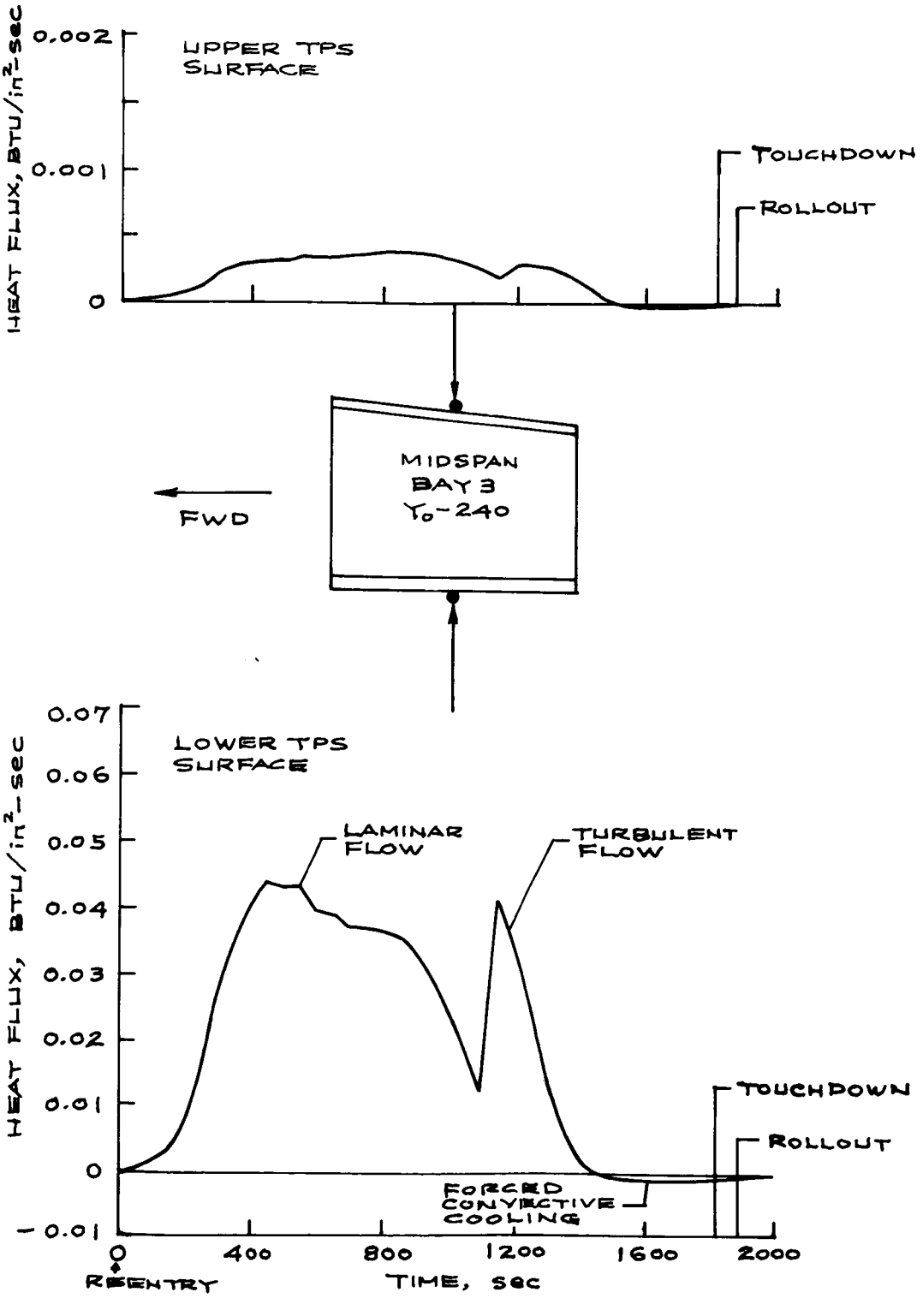


Figure 5. Surface heating rates at midspan bay 3 of orbiter wing; STS-5 flight.

C-2

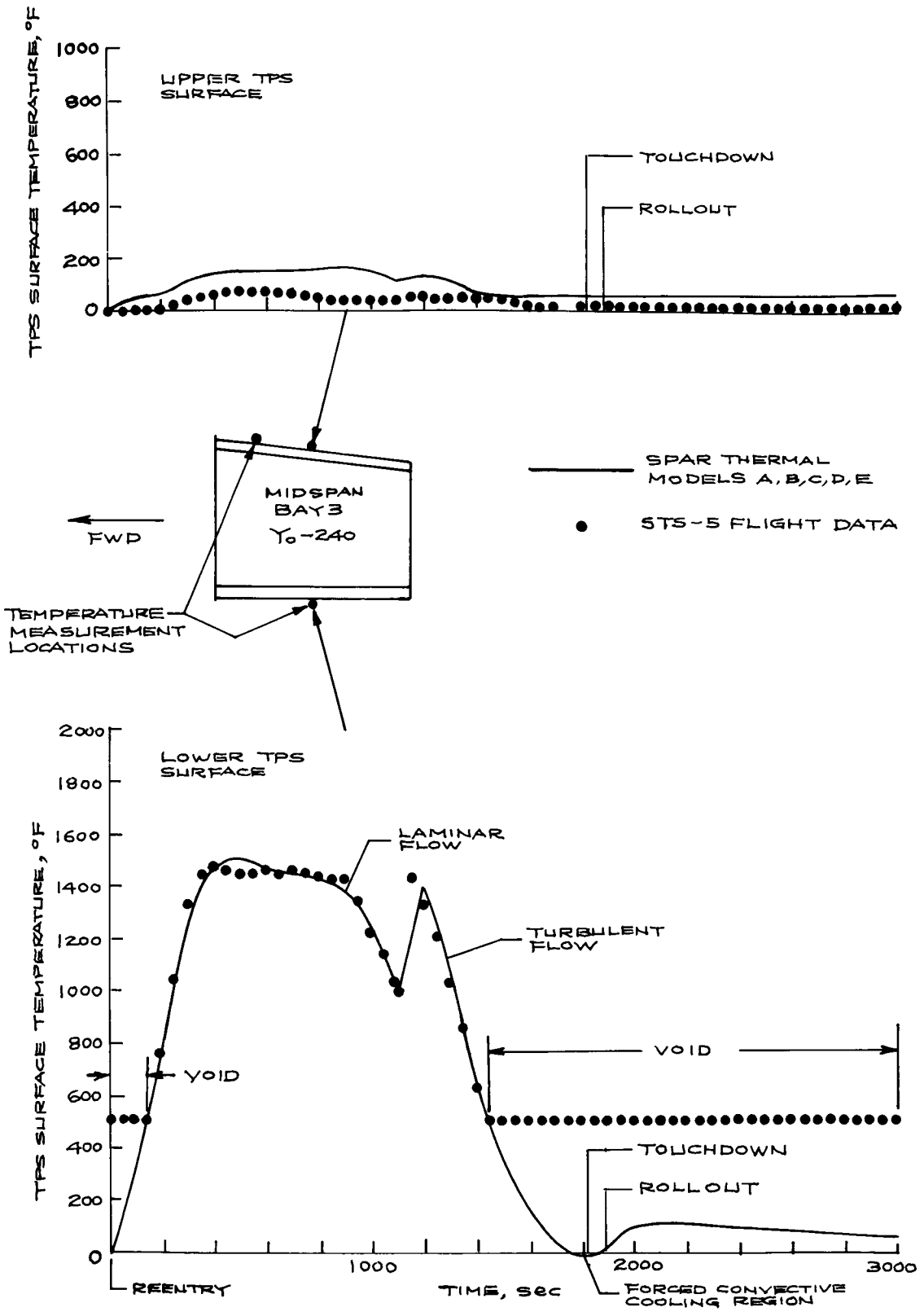


Figure 6. Time histories of TPS surface temperatures calculated using different SPAR thermal models; STS-5 flight.

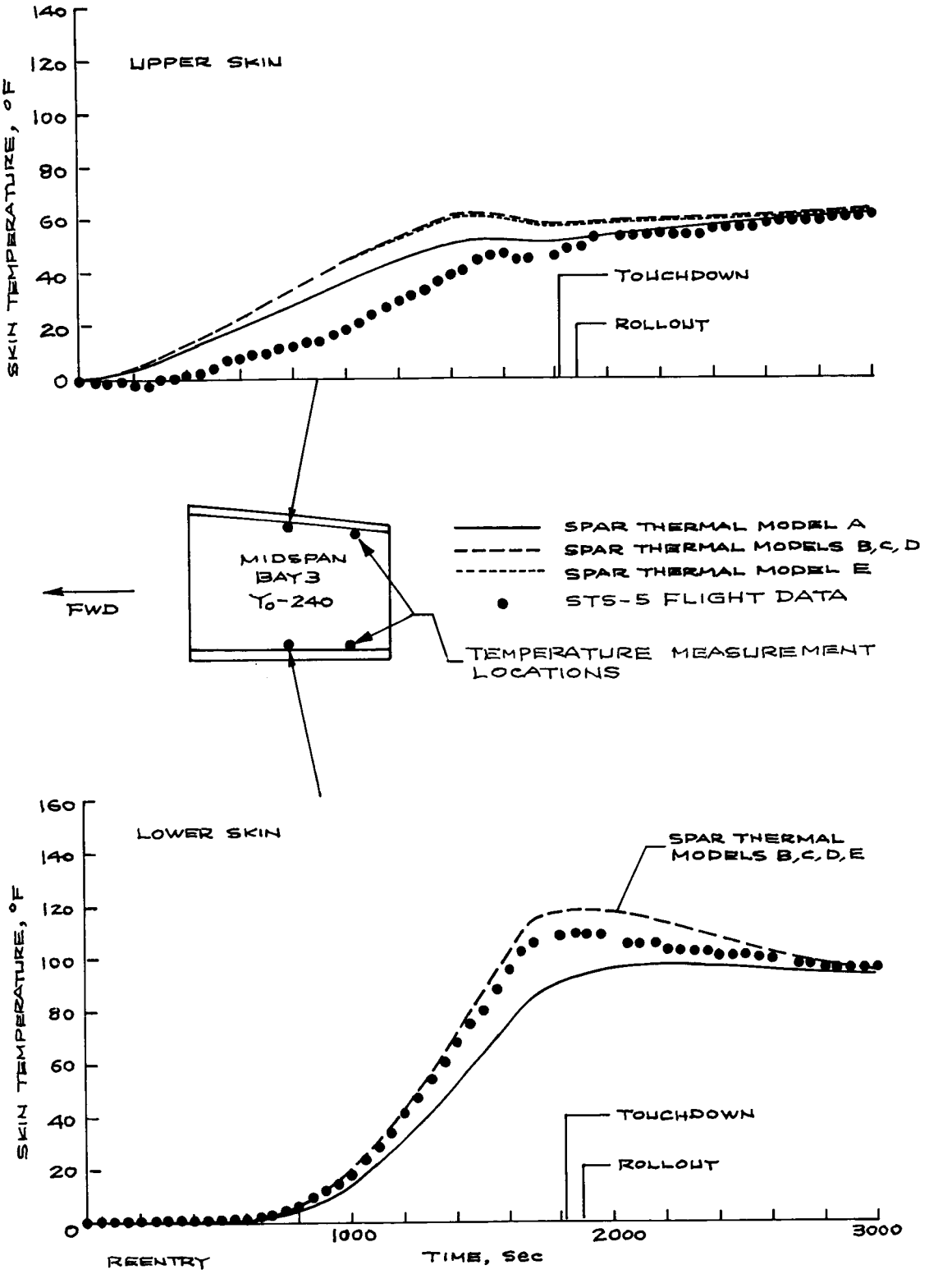
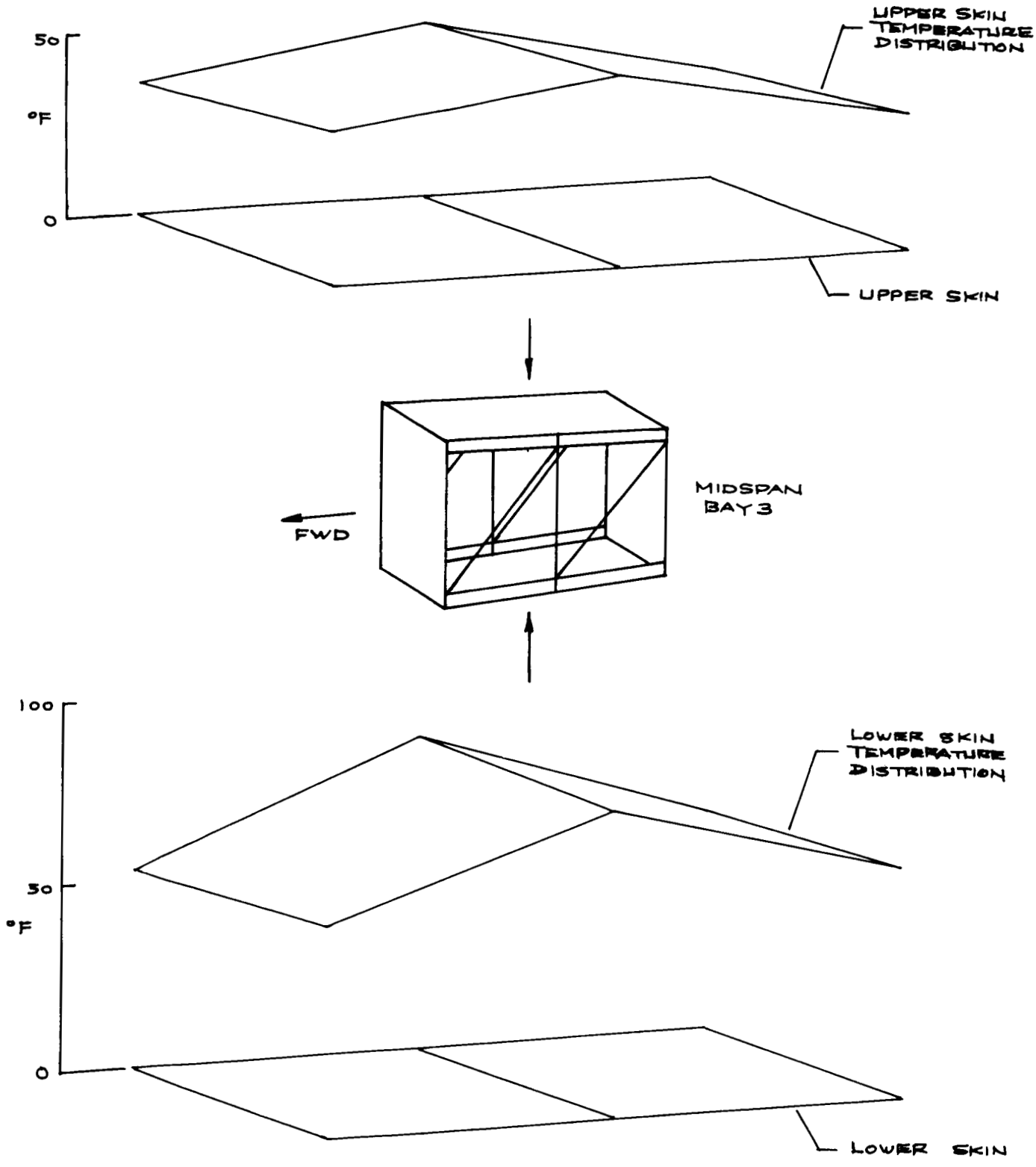
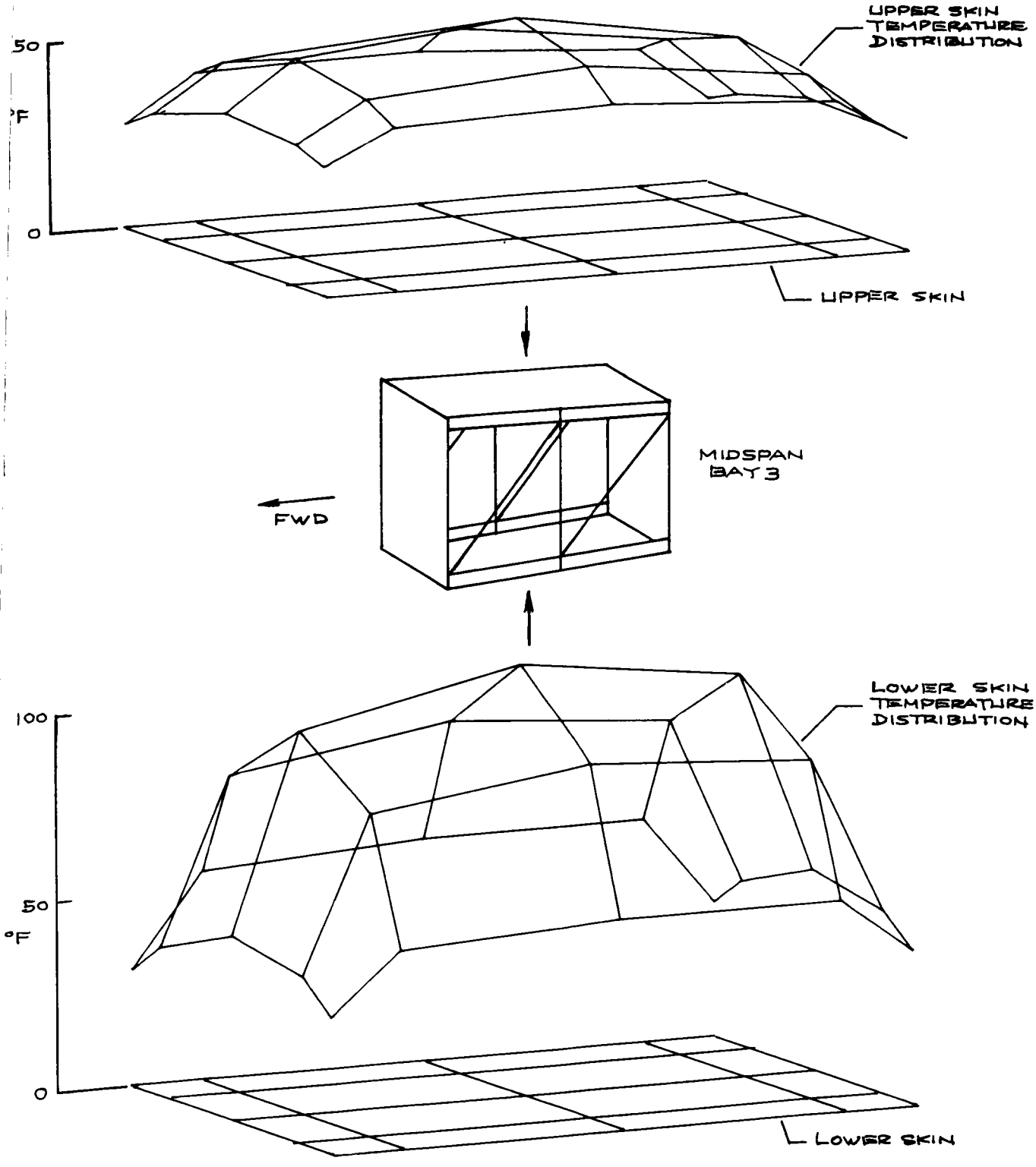


Figure 7. Time histories of orbiter wing skin temperatures calculated using different SPAR thermal models; STS-5 flight.



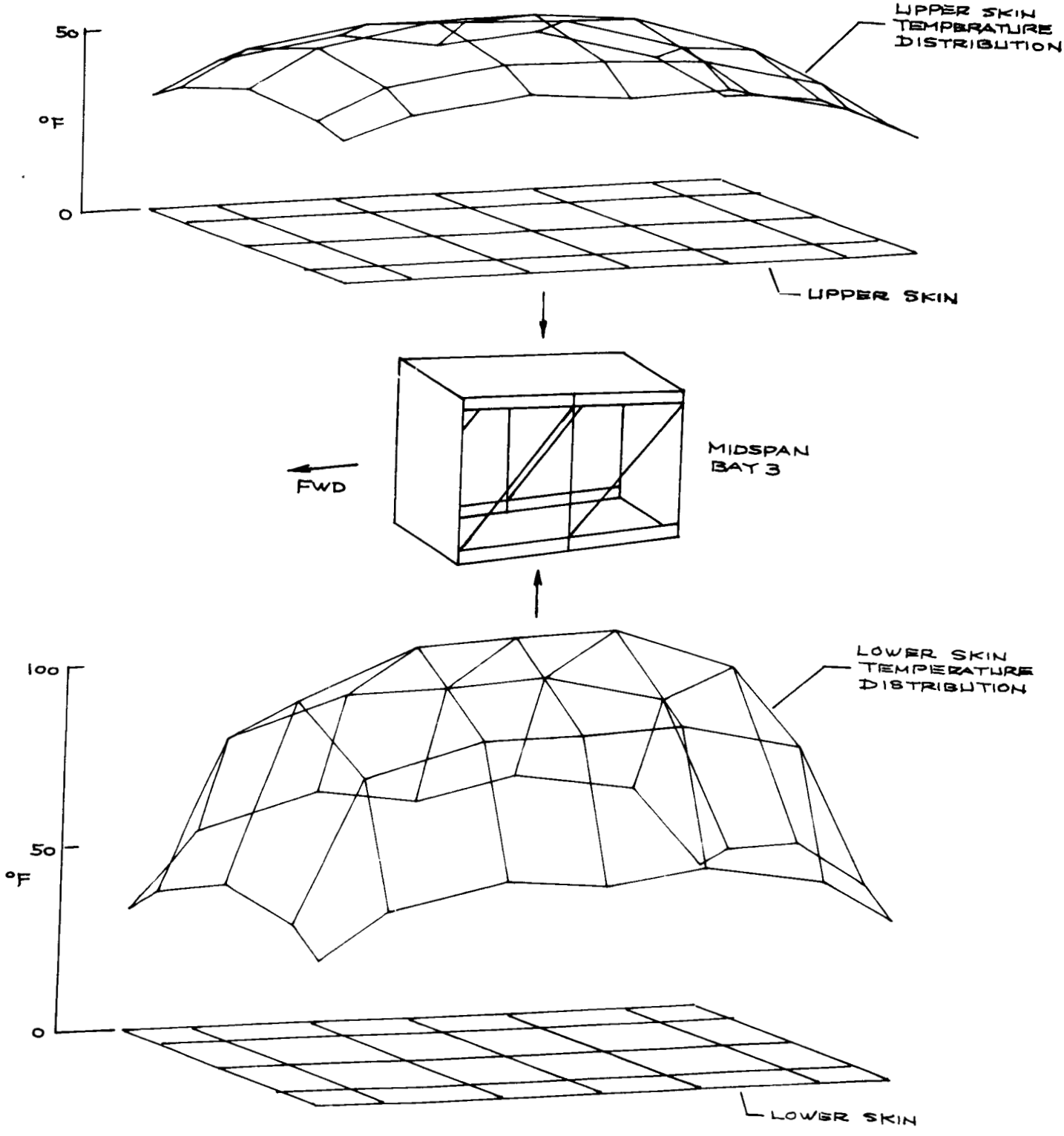
(a) SPAR thermal model A.

Figure 8. Distributions of orbiter wing skin temperatures at midspan bay 3; time = 1700 sec, STS-5 flight.



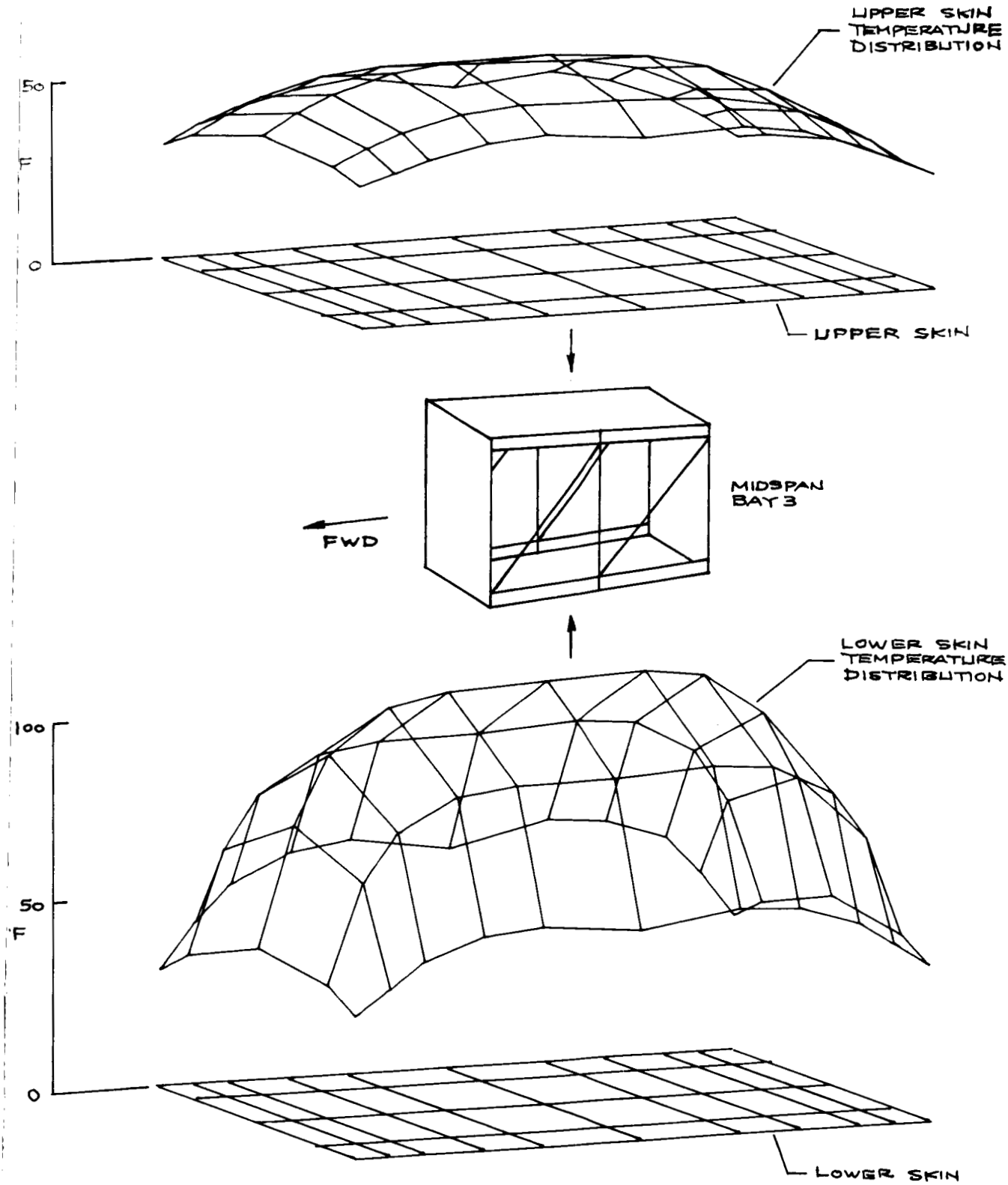
(b) SPAR thermal model B.

Figure 8. Continued.



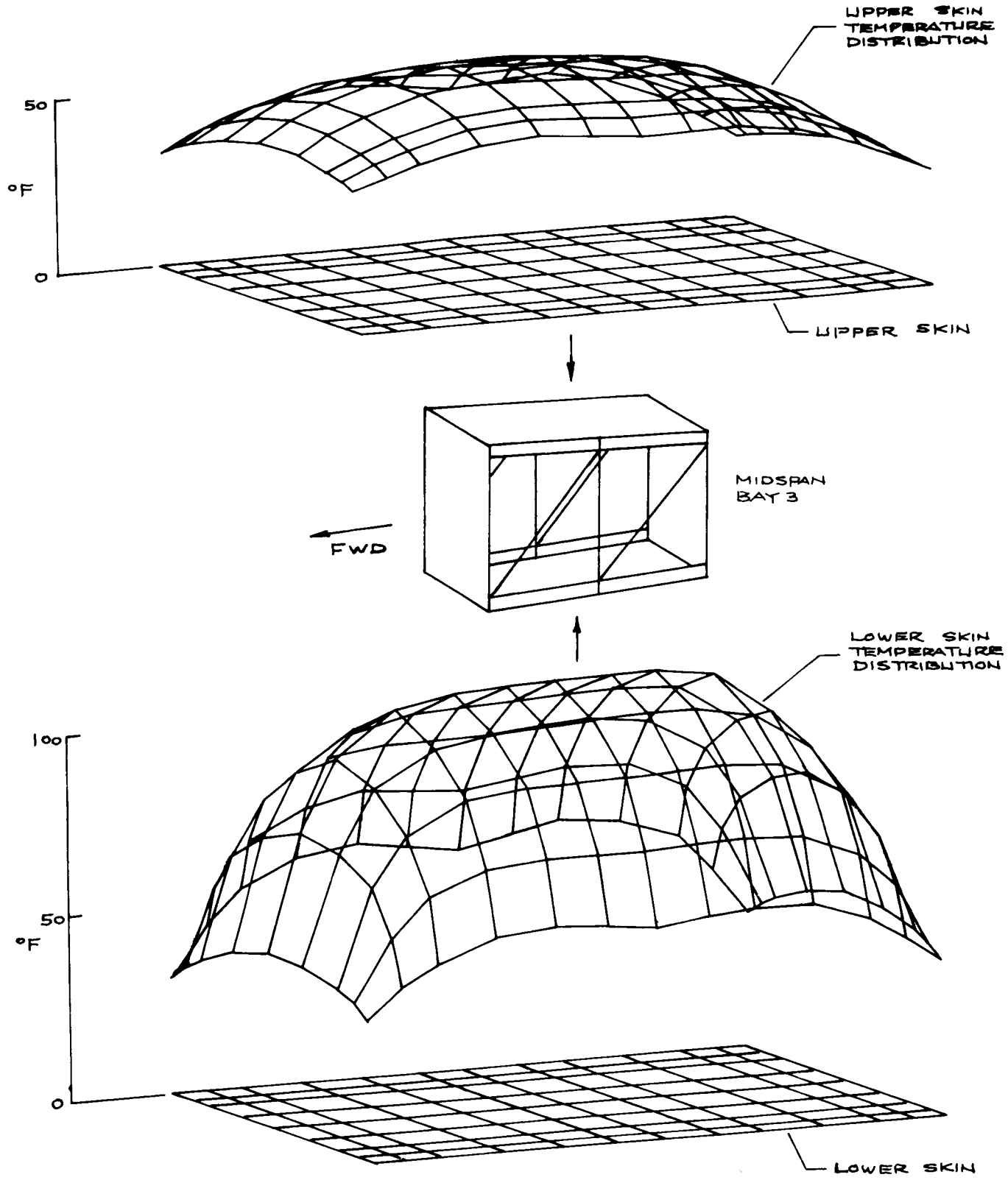
(c) SPAR thermal model C.

Figure 8. Continued.



(d) SPAR thermal model D.

Figure 8. Continued.



(e) SPAR thermal model E.

Figure 8. Concluded.

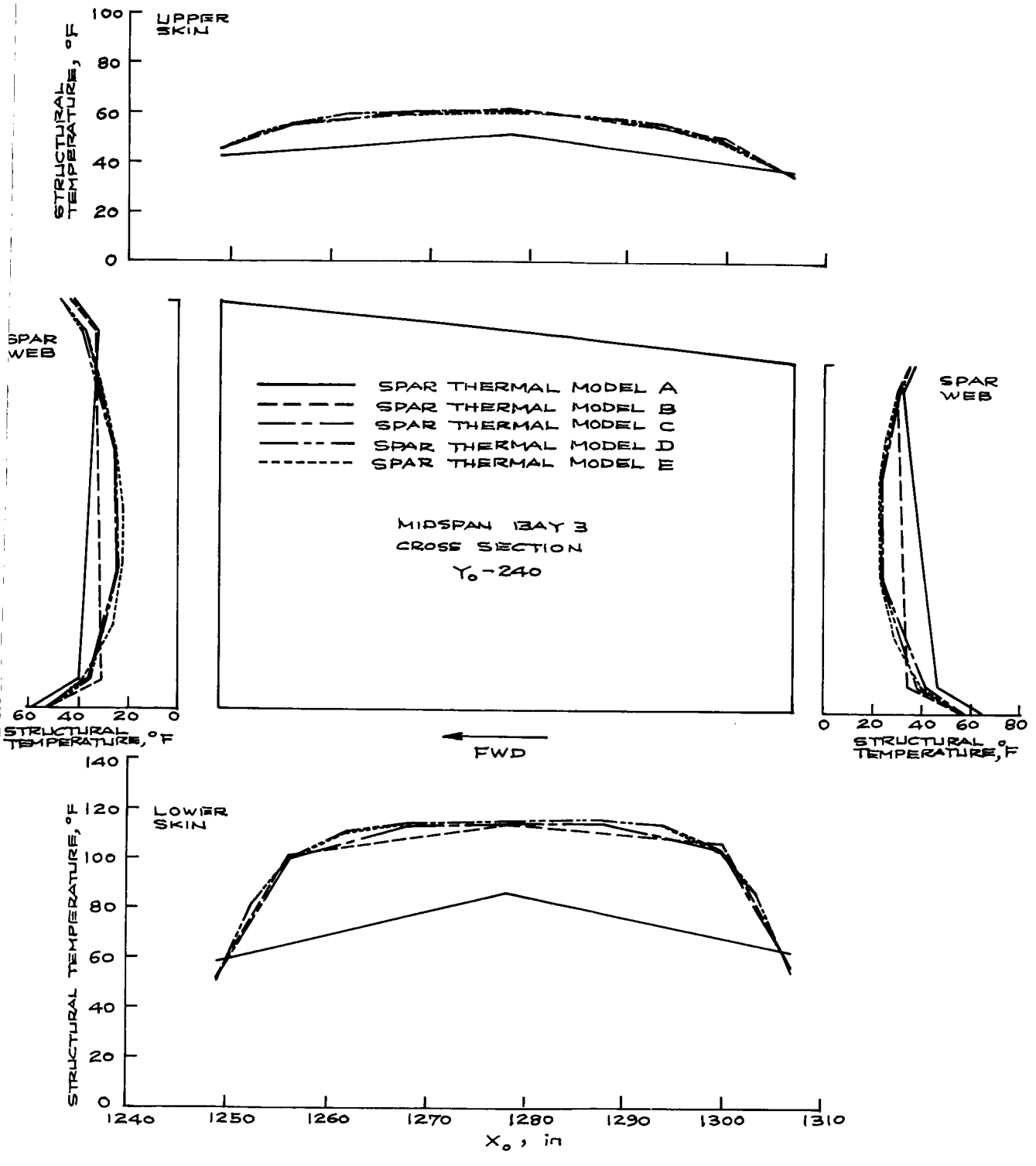


Figure 9. Structural temperature distributions in the Y_0-240 plane of orbiter wing midspan bay 3 calculated using different SPAR thermal models; time = 1700 sec, STS-5 flight.

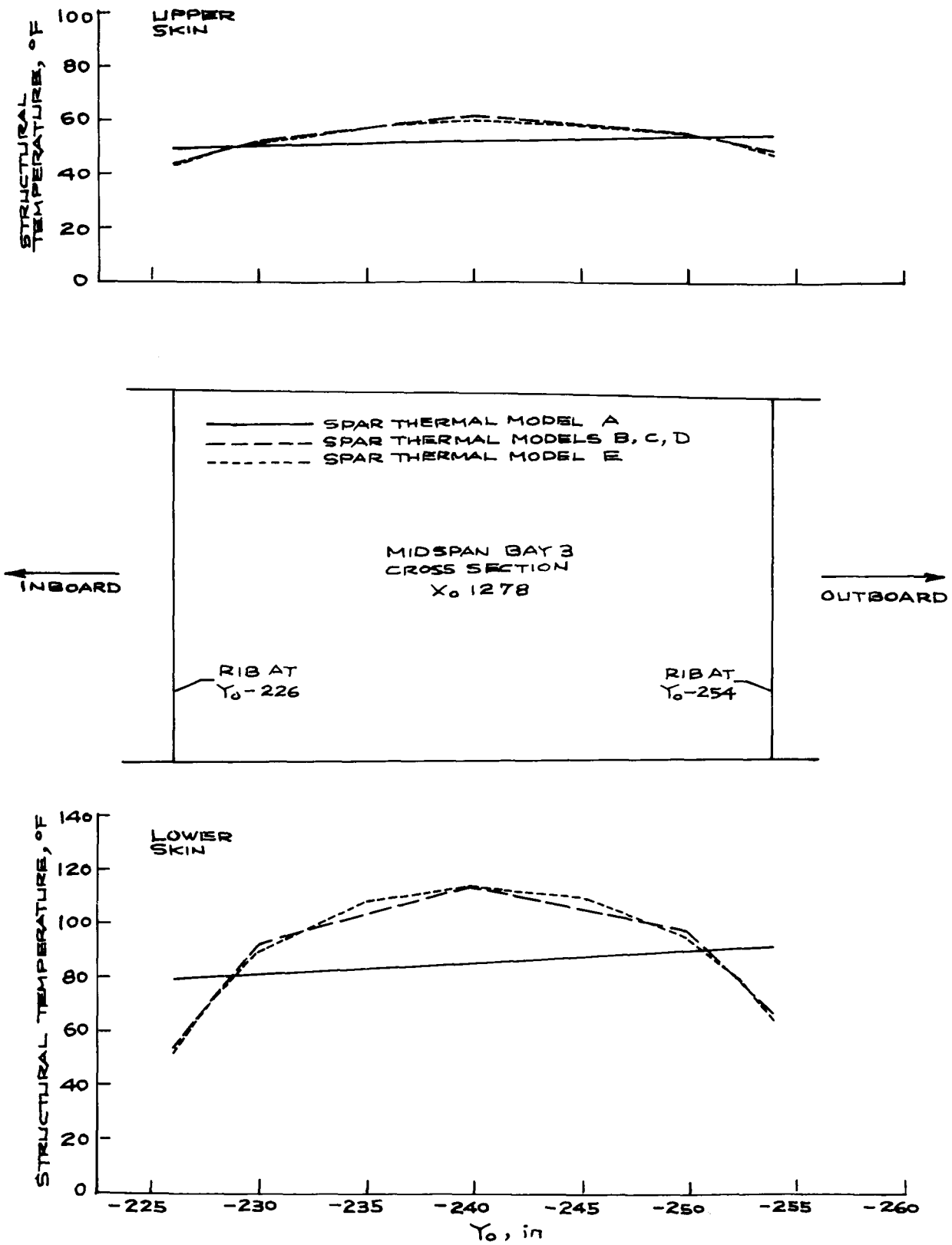


Figure 10. Spanwise distributions in the $X_0$1278 plane of structural temperatures in orbiter wing midspan bay 3 calculated using different SPAR thermal models; time = 1700 sec, STS-5 flight.

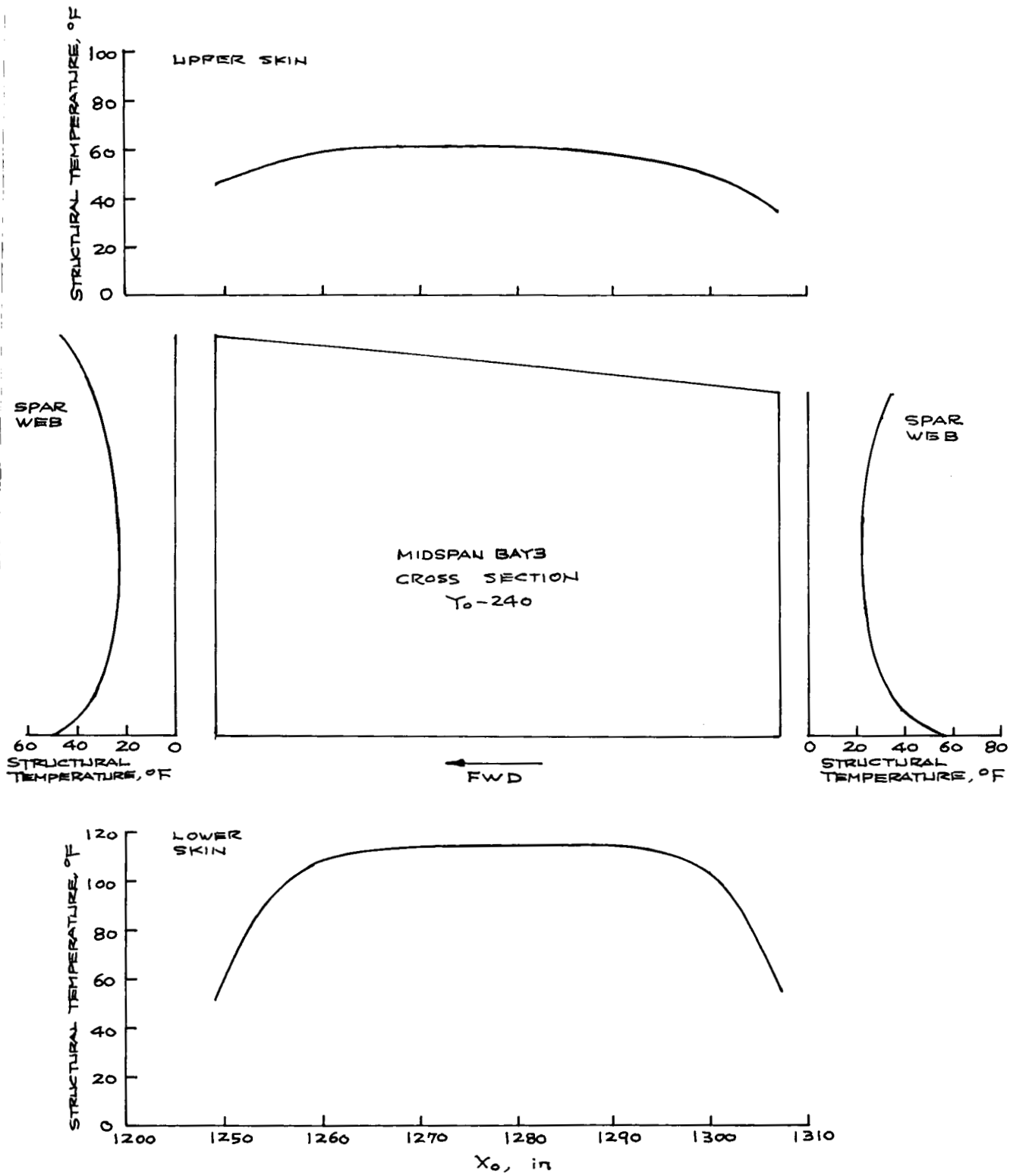


Figure 11. Continuous distributions in the Y_0-240 plane of structural temperatures based on SPAR thermal model E; time = 1700 sec, STS-5 flight.

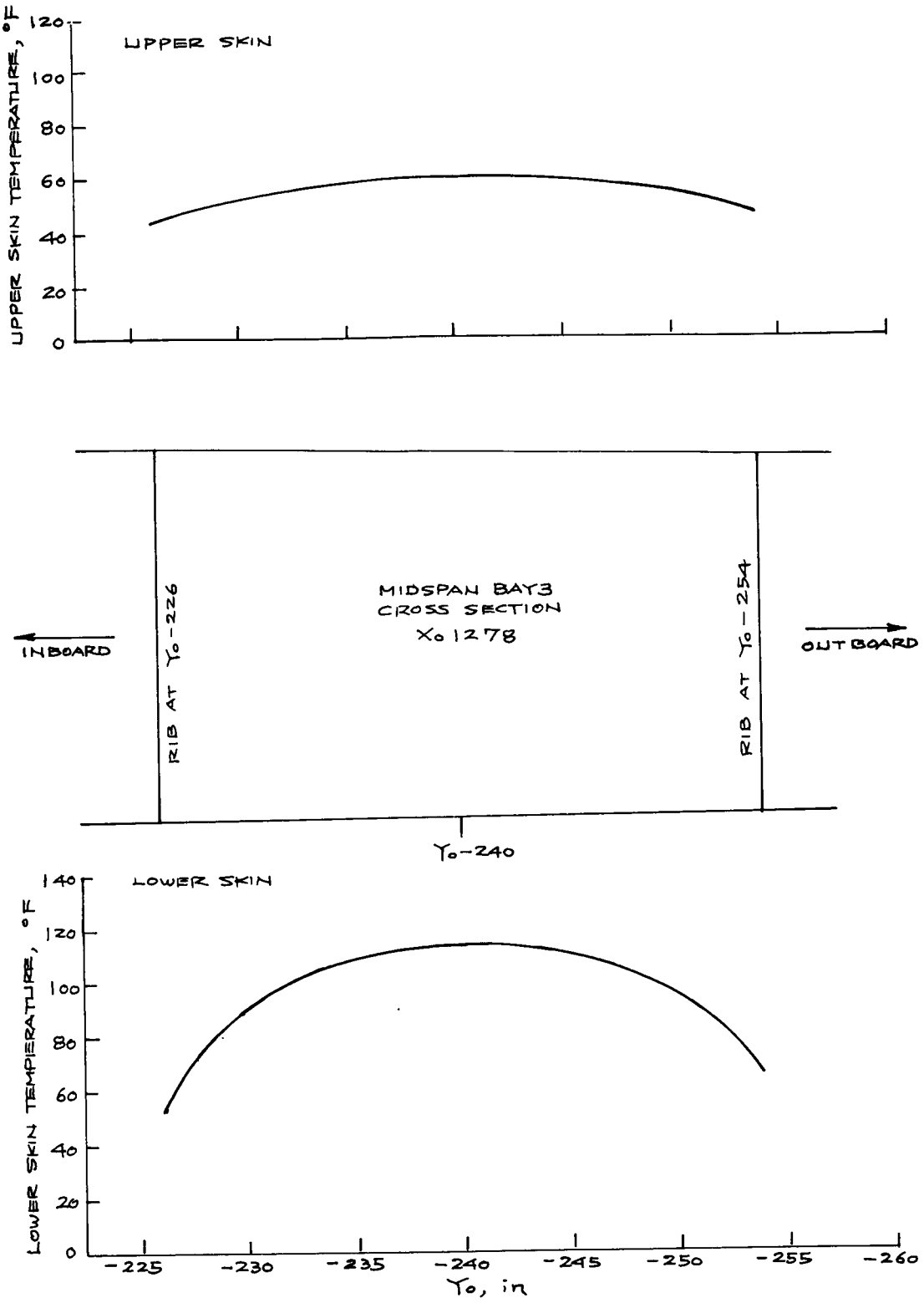
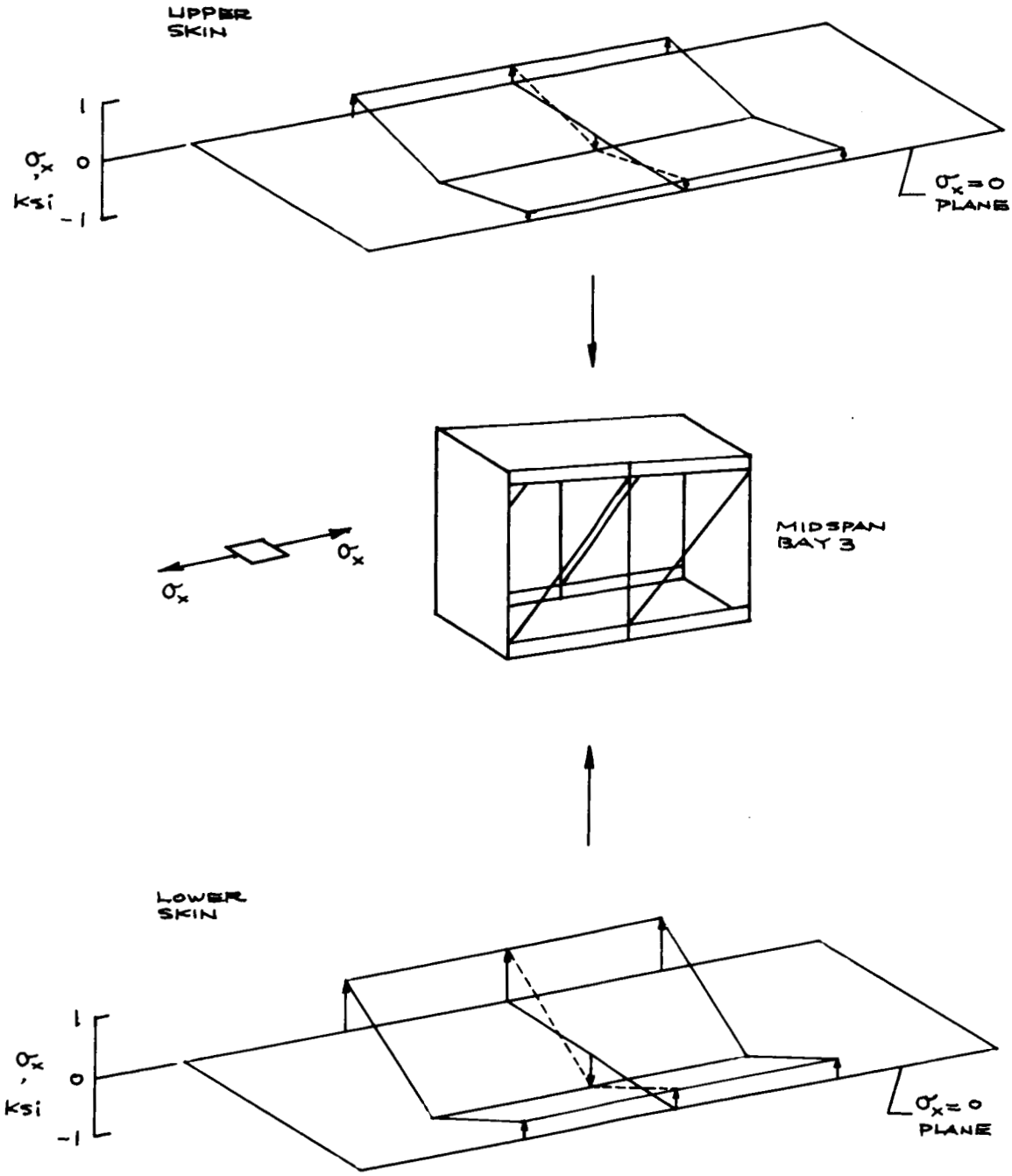
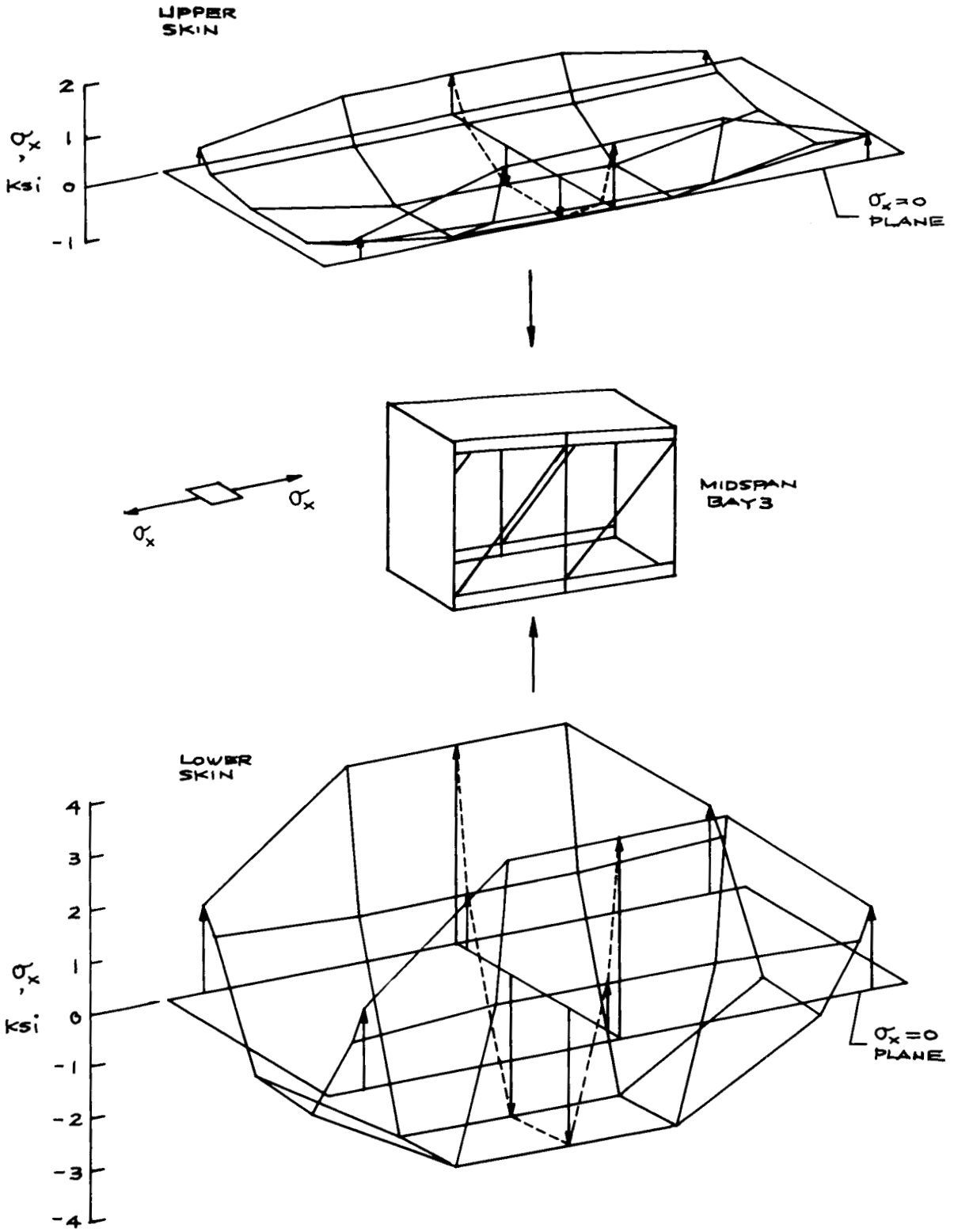


Figure 12. Continuous distributions of wing skin temperatures in the $X_0 1278$ plane based on thermal model E; time = 1700 sec, STS-5 flight.



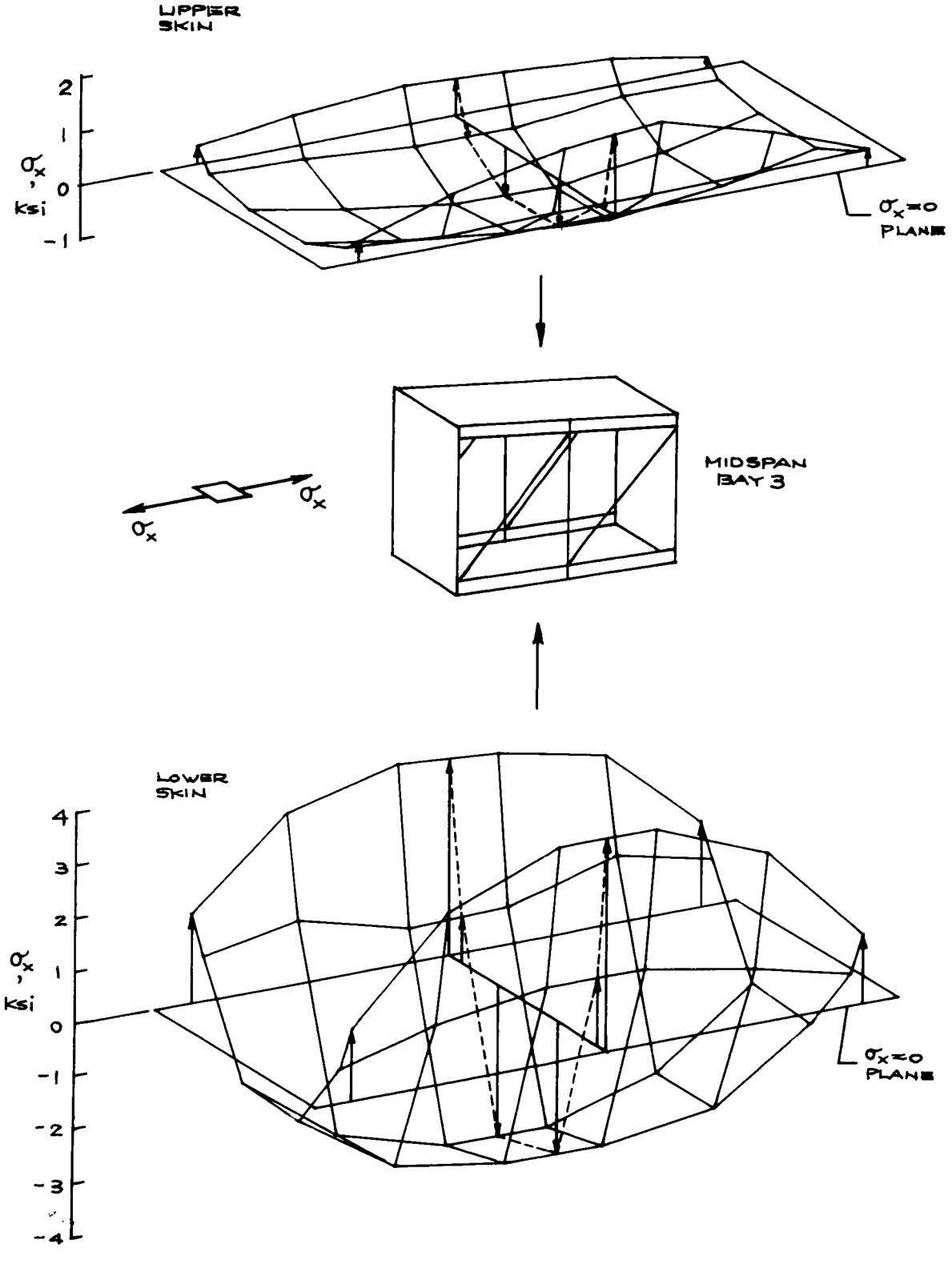
(a) NASTRAN structural model A.

Figure 13. Distributions of chordwise stress σ_x in orbiter wing skins at midspan bay 3; time = 1700 sec, STS-5 flight.



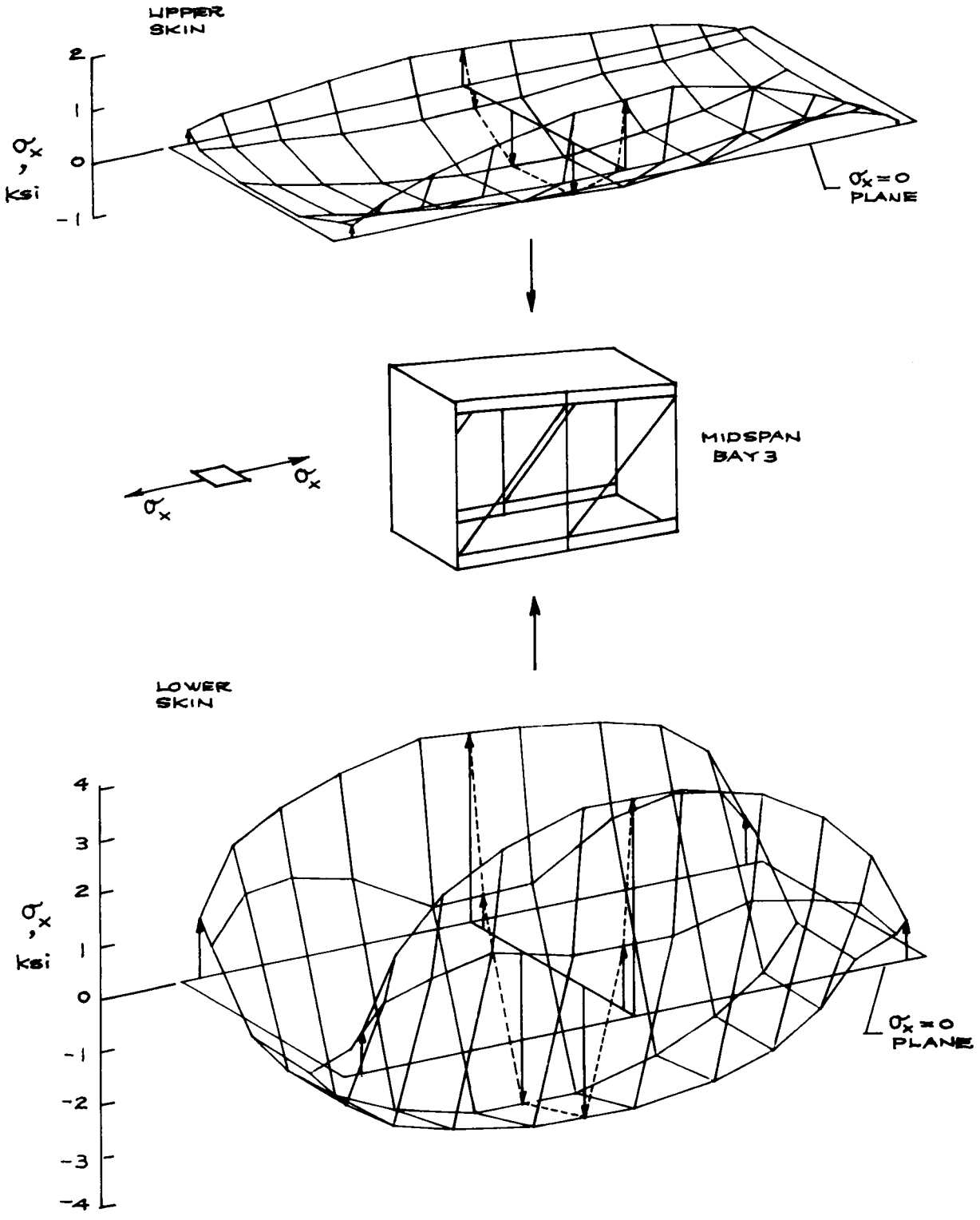
(b) NASTRAN structural model B.

Figure 13. Continued.



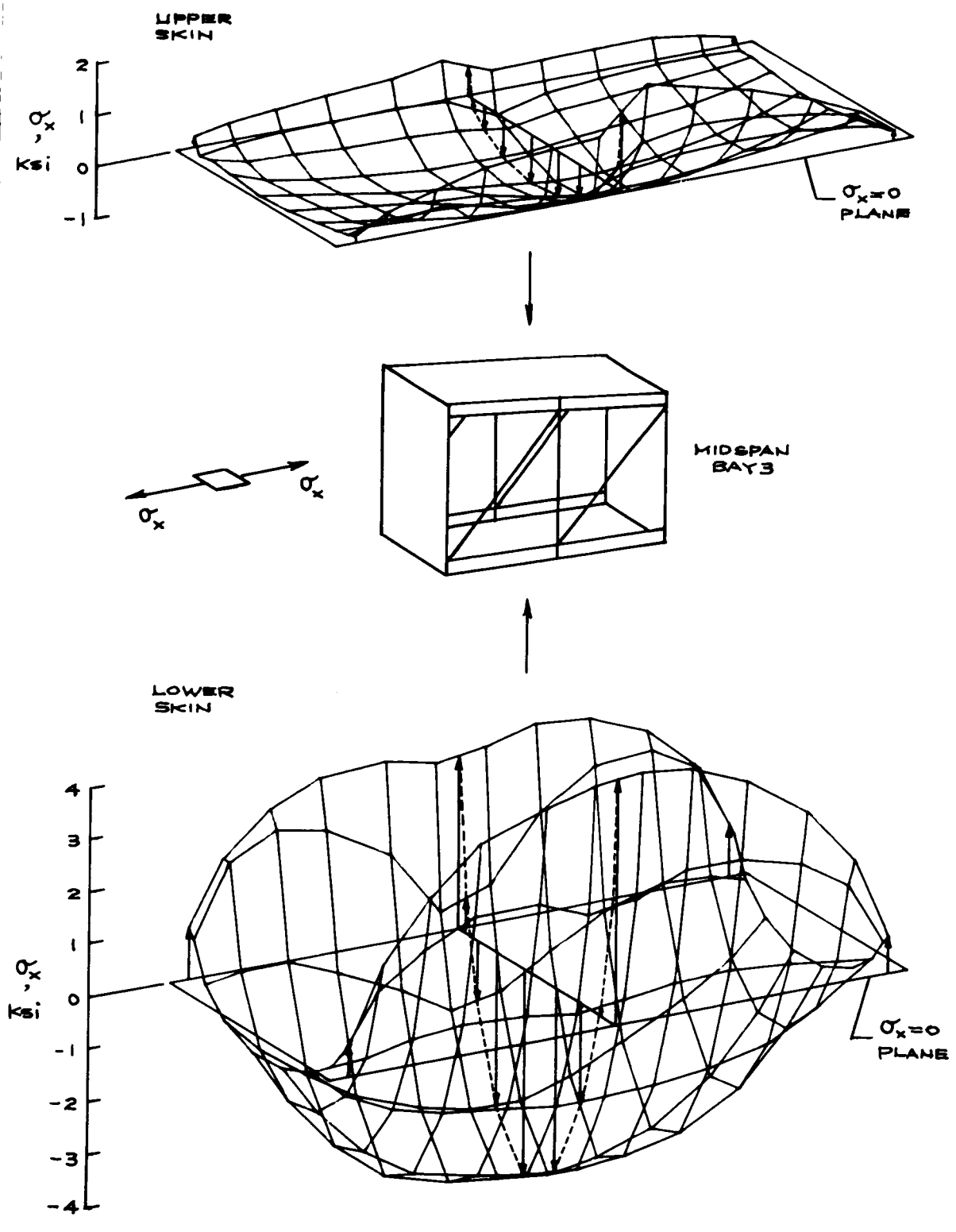
(c) NASTRAN structural model C.

Figure 13. Continued.



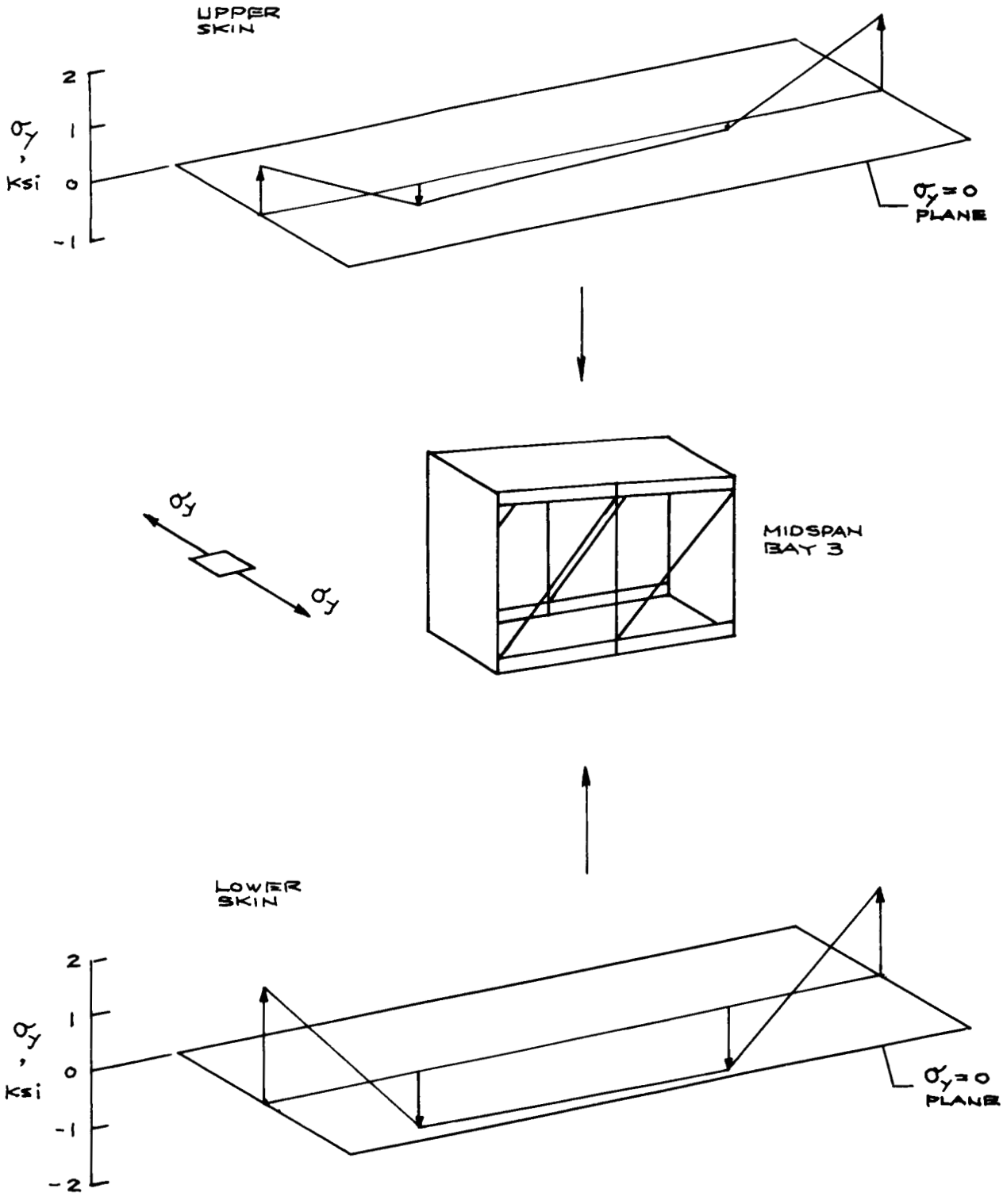
(d) NASTRAN structural model D.

Figure 13. Continued.



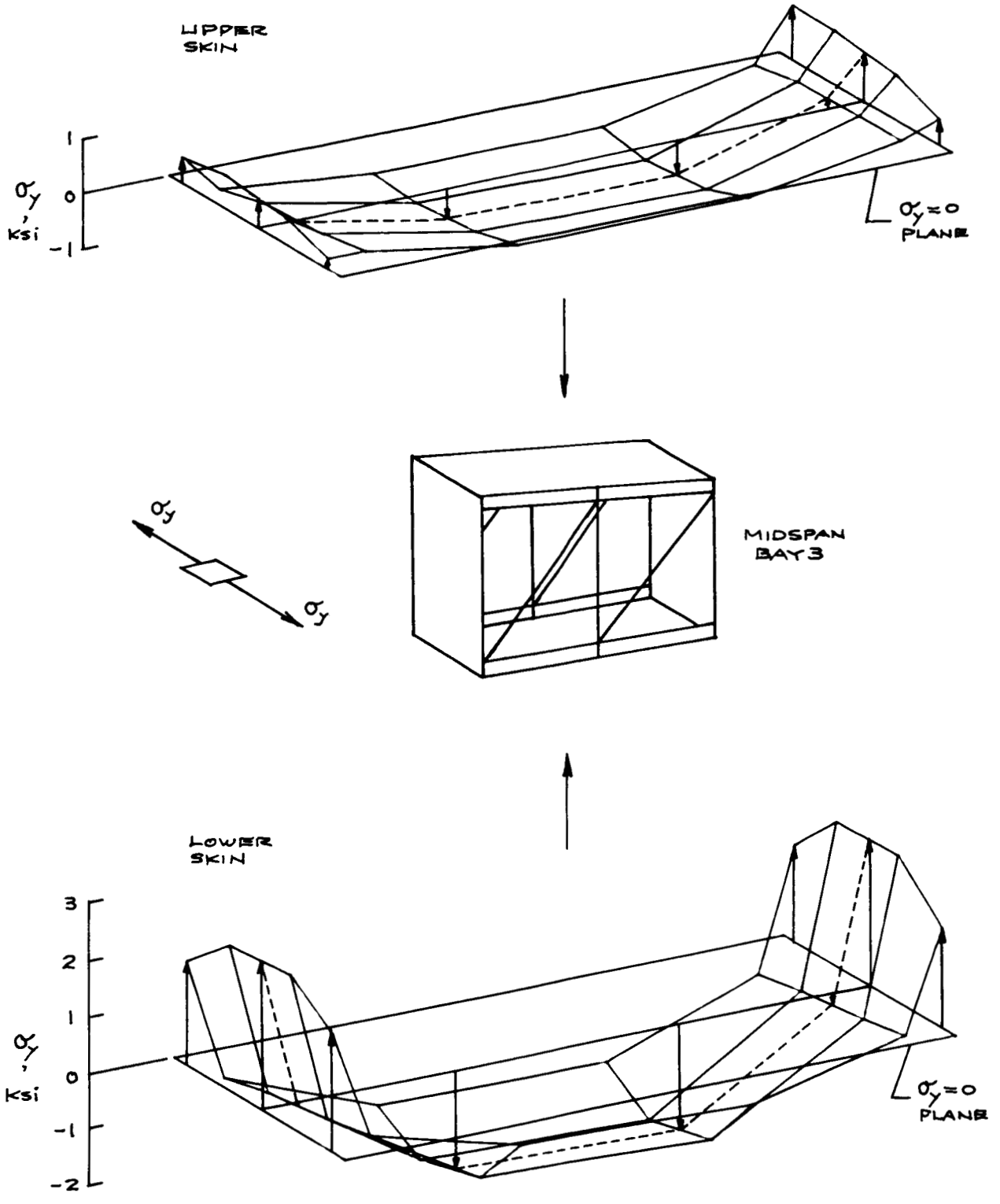
(e) NASTRAN structural model E.

Figure 13. Concluded.



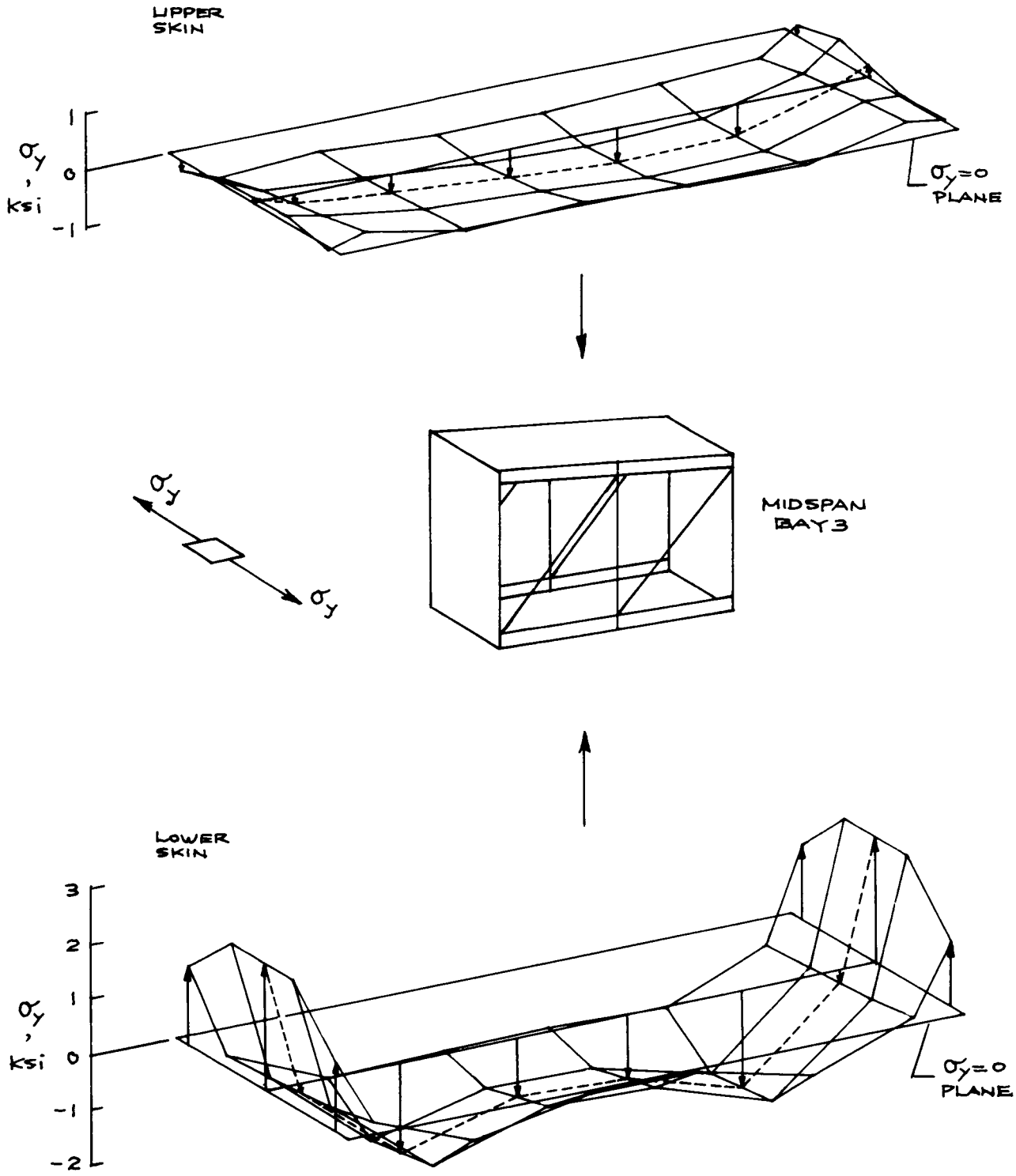
(a) NASTRAN structural model A.

Figure 14. Distributions of spanwise stress σ_y in orbiter wing skins at midspan bay 3; time = 1700 sec, STS-5 flight.



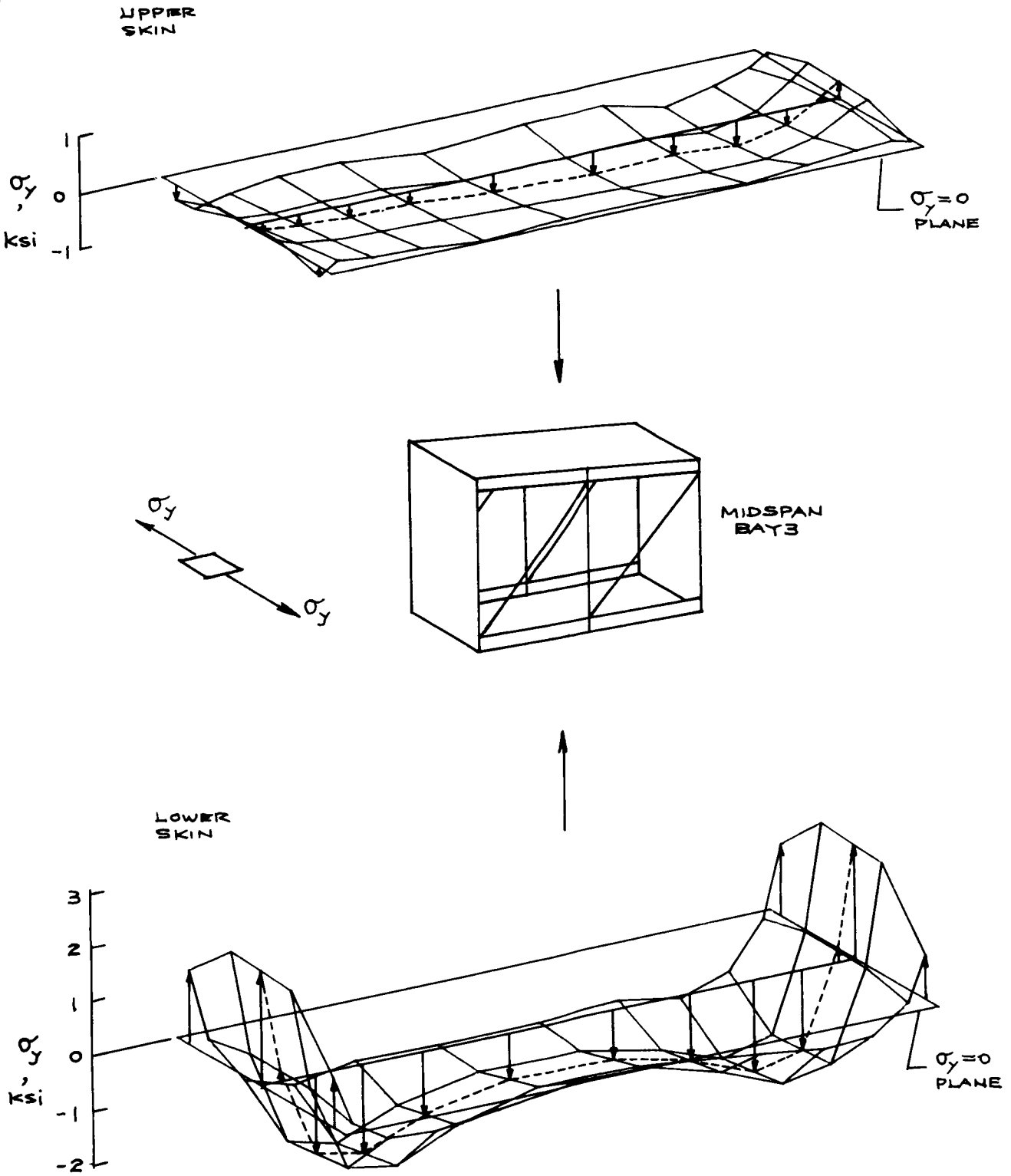
(b) NASTRAN structural model B.

Figure 14. Continued.



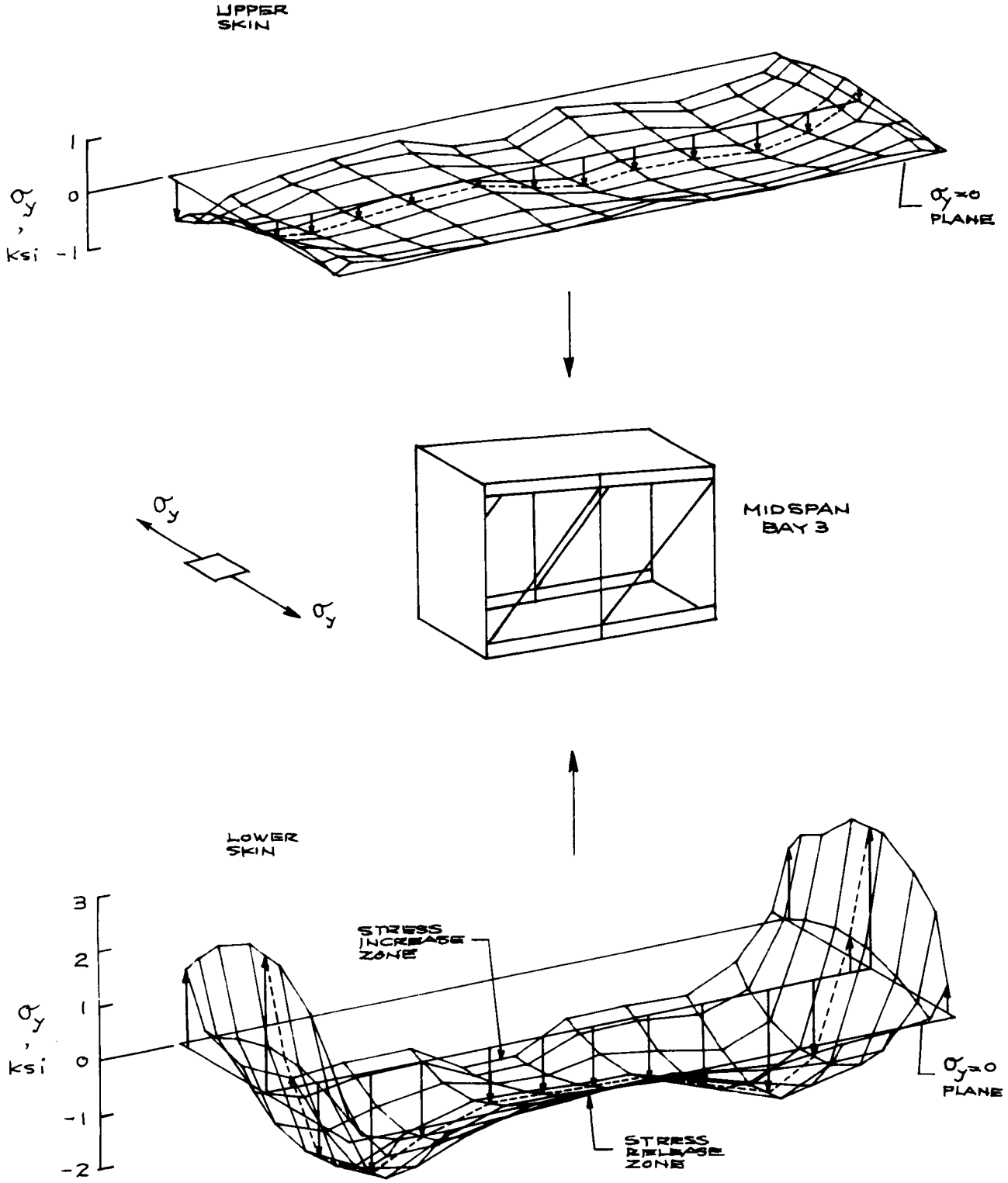
(c) NASTRAN structural model C.

Figure 14. Continued.



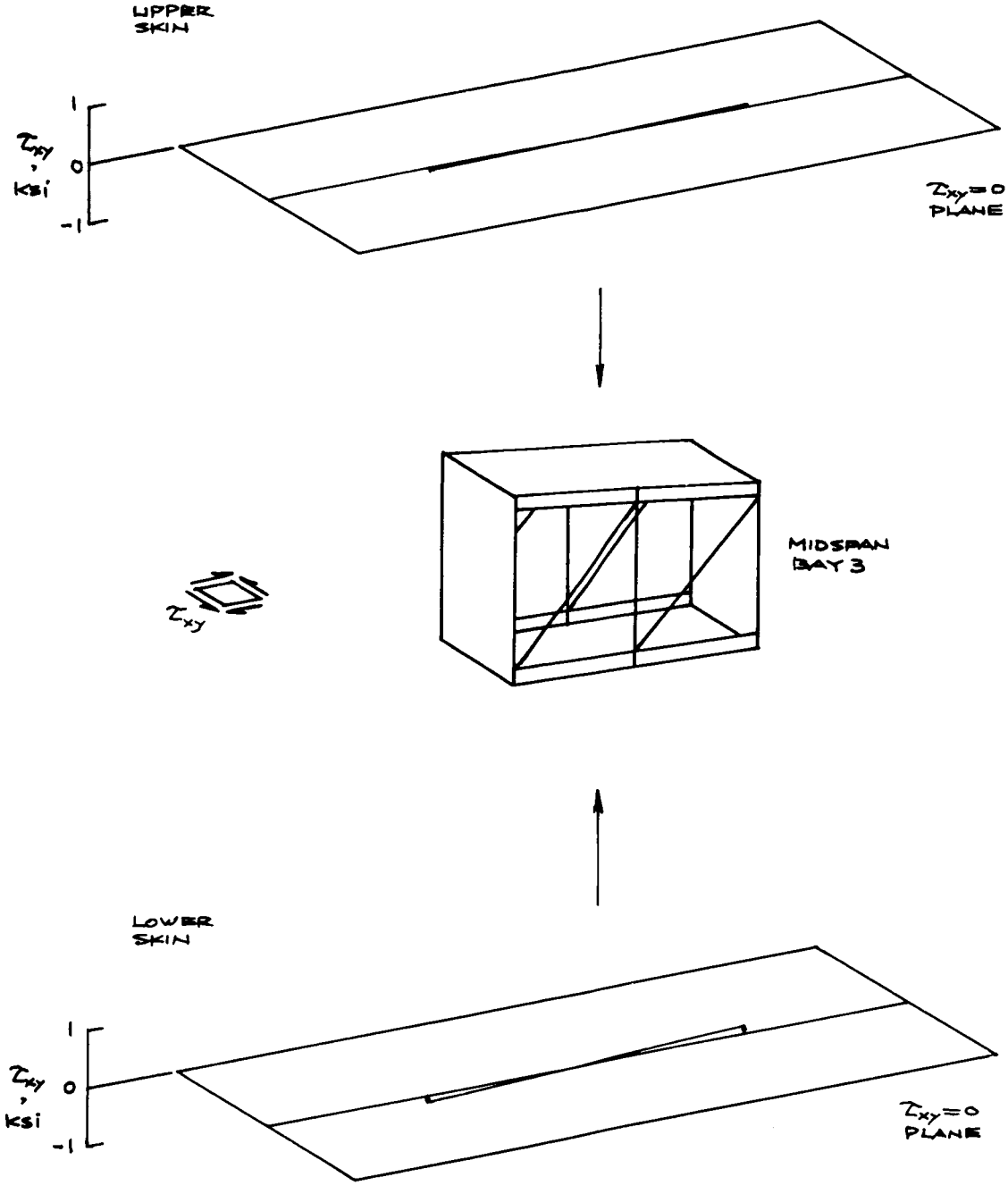
(d) NASTRAN structural model D.

Figure 14. Continued.



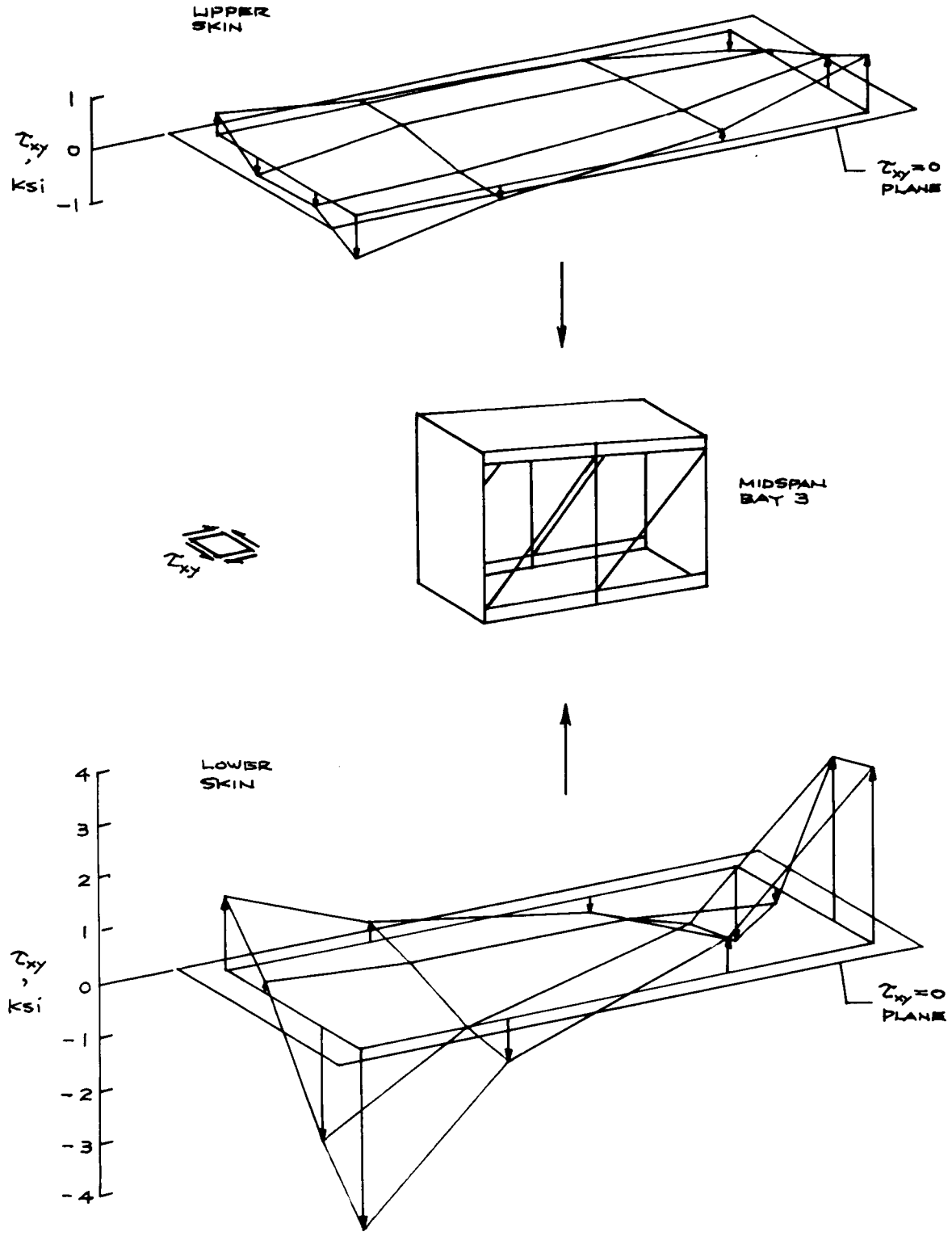
(e) NASTRAN structural model E.

Figure 14. Concluded.



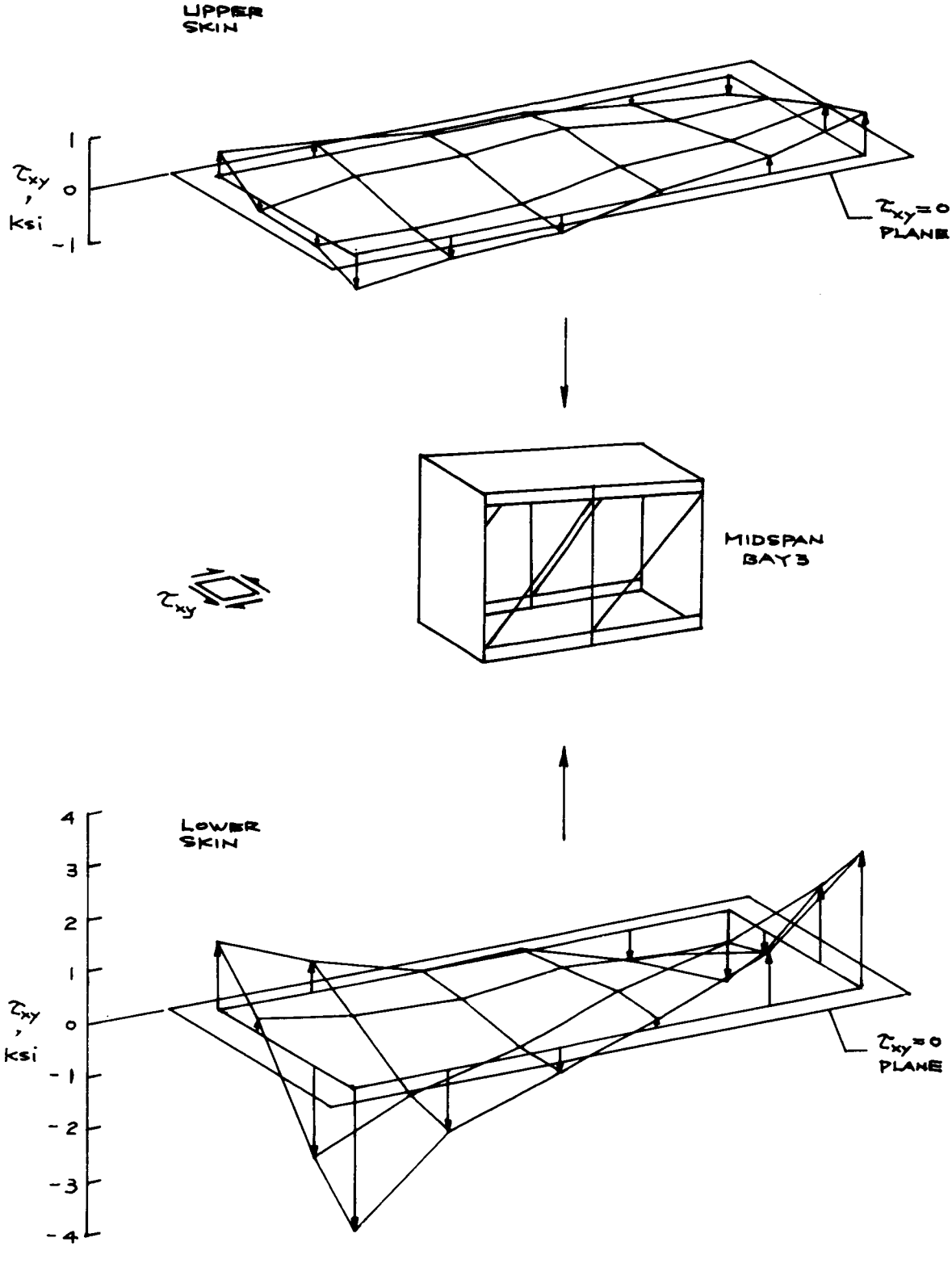
(a) NASTRAN structural model A.

Figure 15. Distributions in the Y_0-240 plane of shear stress τ_{xy} in orbiter wing skins at midspan bay 3; time = 1700 sec, STS-5 flight.



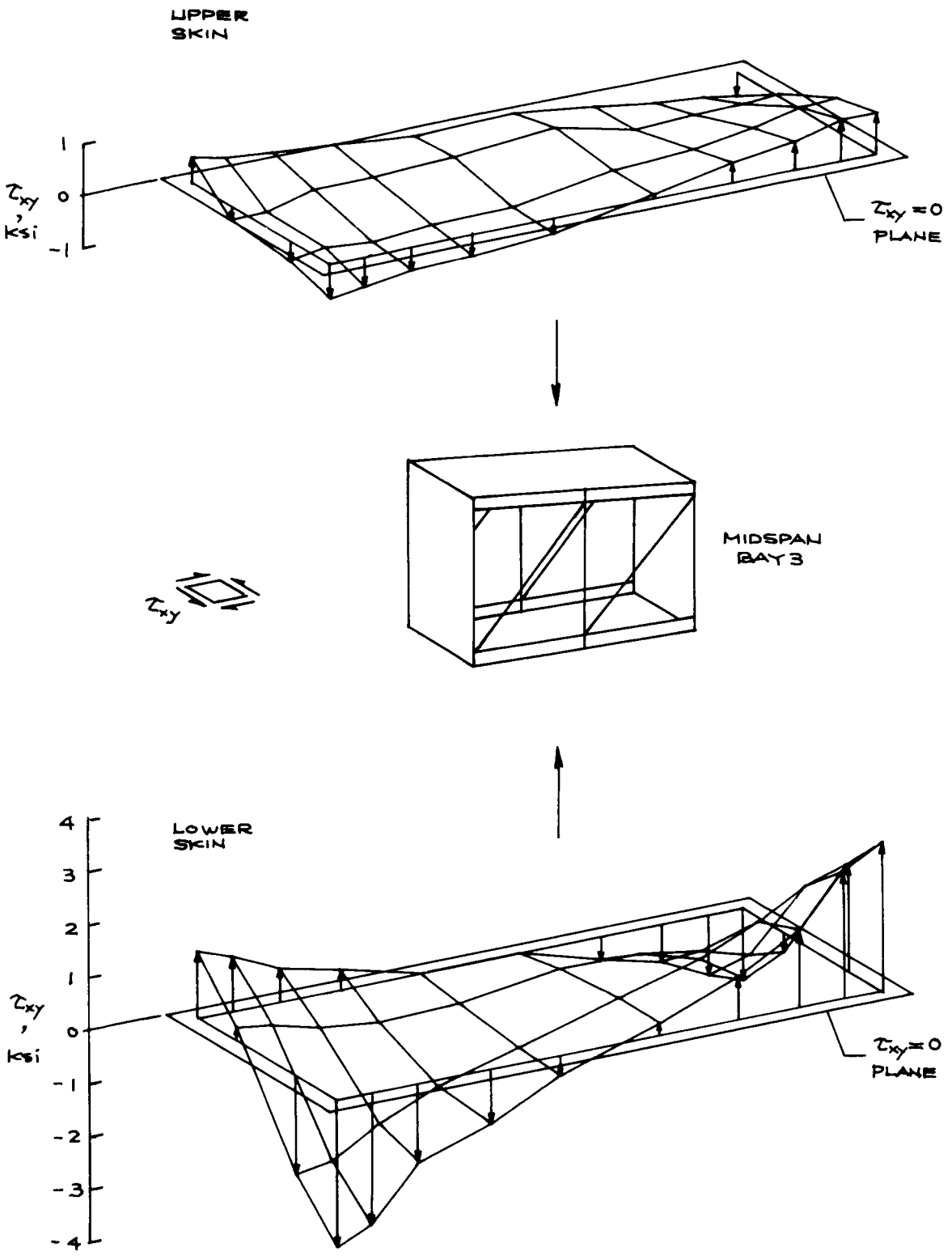
(b) NASTRAN structural model B.

Figure 15. Continued.



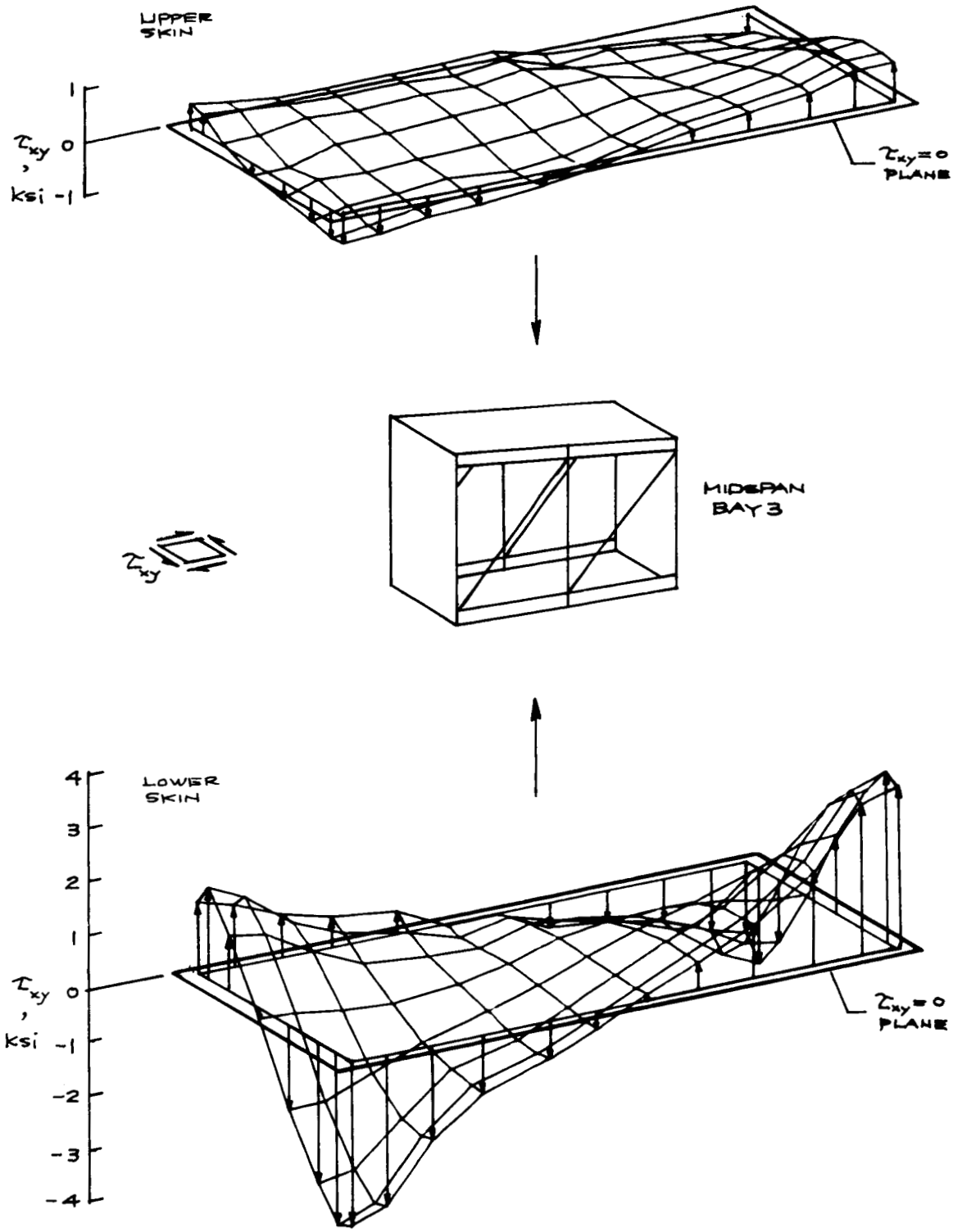
(c) NASTRAN structural model C.

Figure 15. Continued.



(d) NASTRAN structural model D.

Figure 15. Continued.



(e) NASTRAN structural model E.

Figure 15. Concluded.

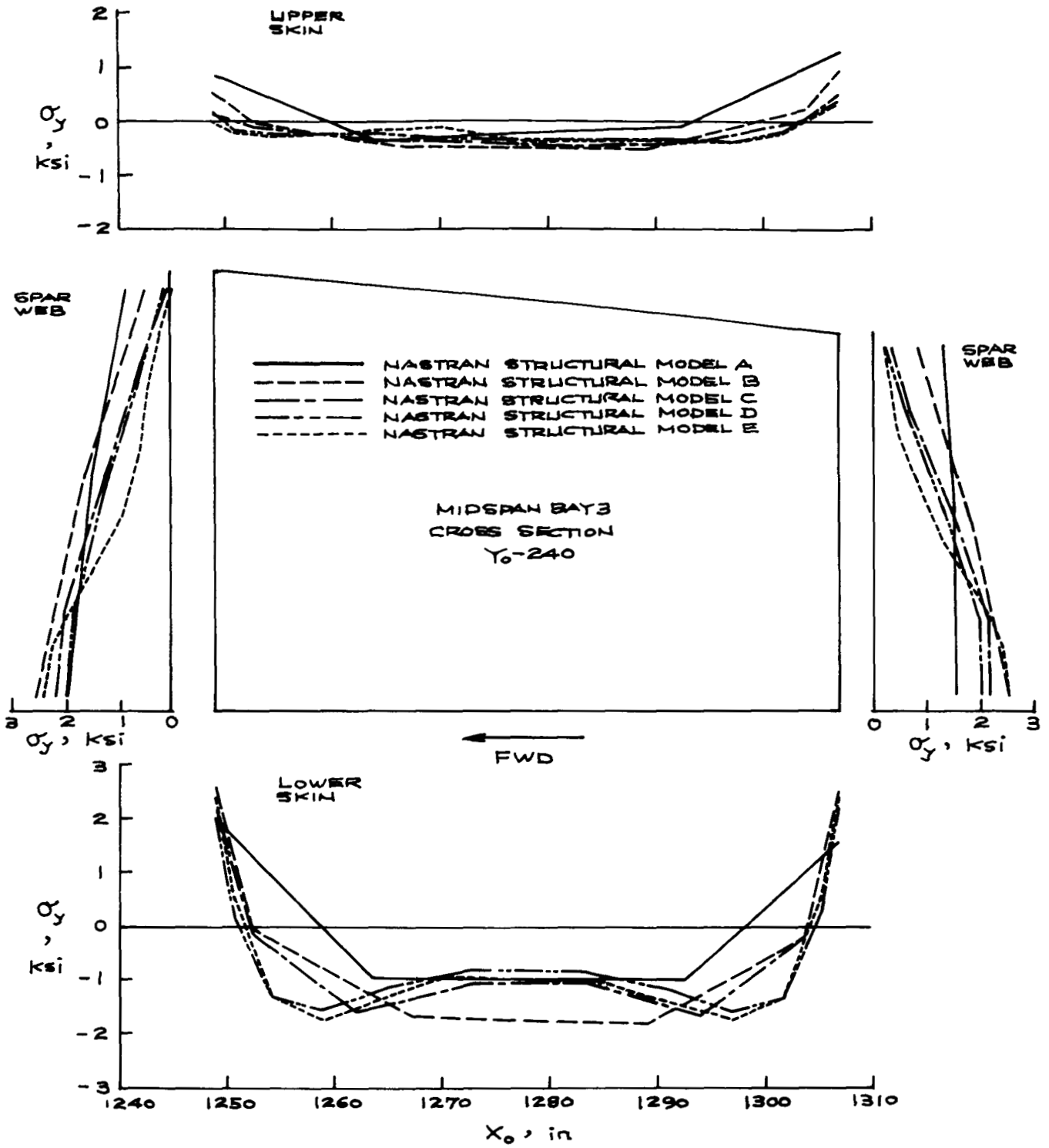


Figure 16. Distributions of spanwise stress σ_y in orbiter wing midspan bay 3 calculated using different NASTRAN structural models; time = 1700 sec, STS-5 flight.

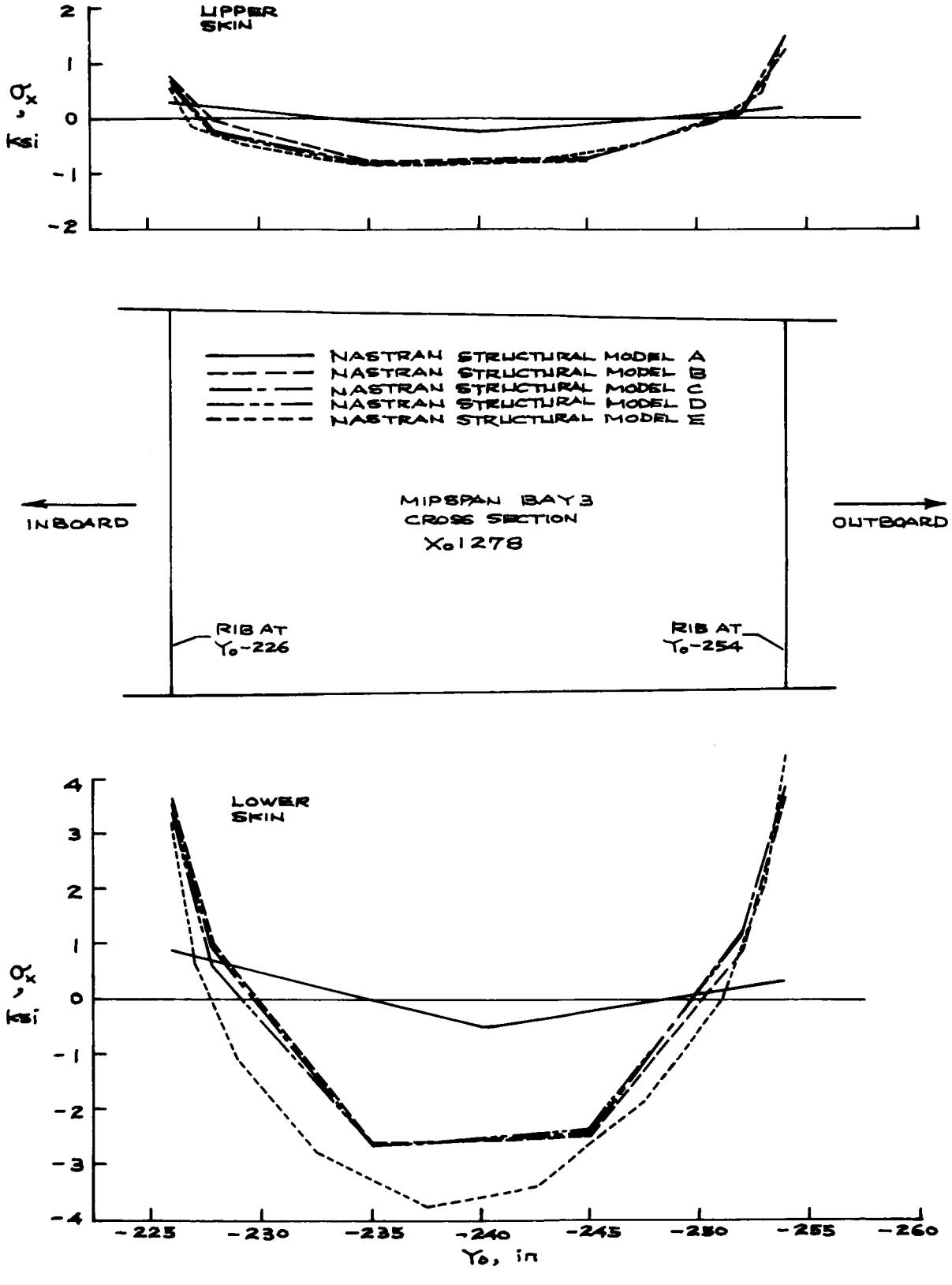


Figure 17. Distributions in the $X_0 1278$ plane of chordwise stress σ_x in orbiter wing midspan bay 3 calculated using different NASTRAN structural models; time = 1700 sec, STS-5 flight.

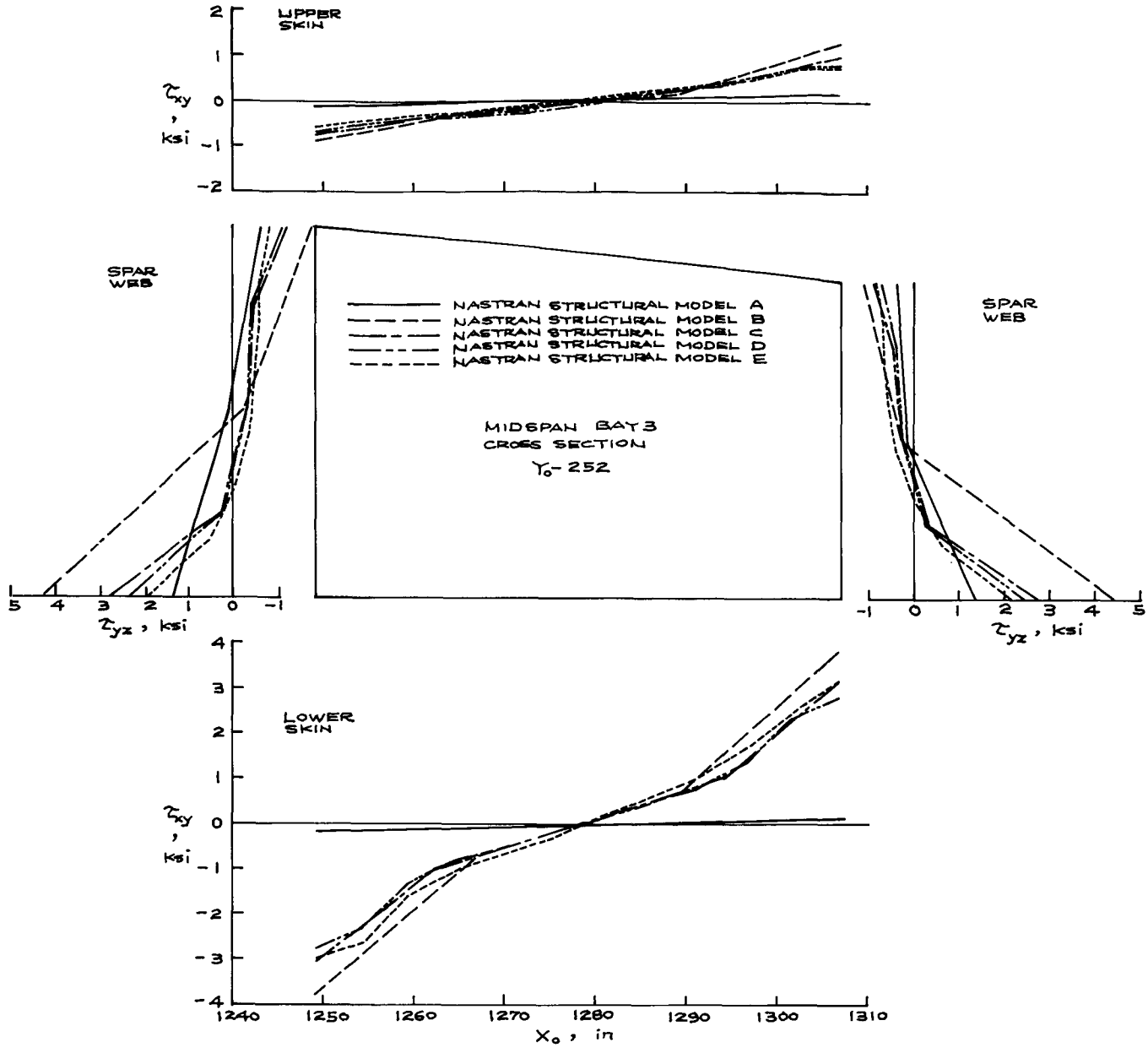


Figure 18. Distributions in the Y_0-252 plane of shear stresses τ_{xy} and τ_{yz} in orbiter wing midspan bay 3 calculated using different NASTRAN structural models; time = 1700 sec, STS-5 flight.

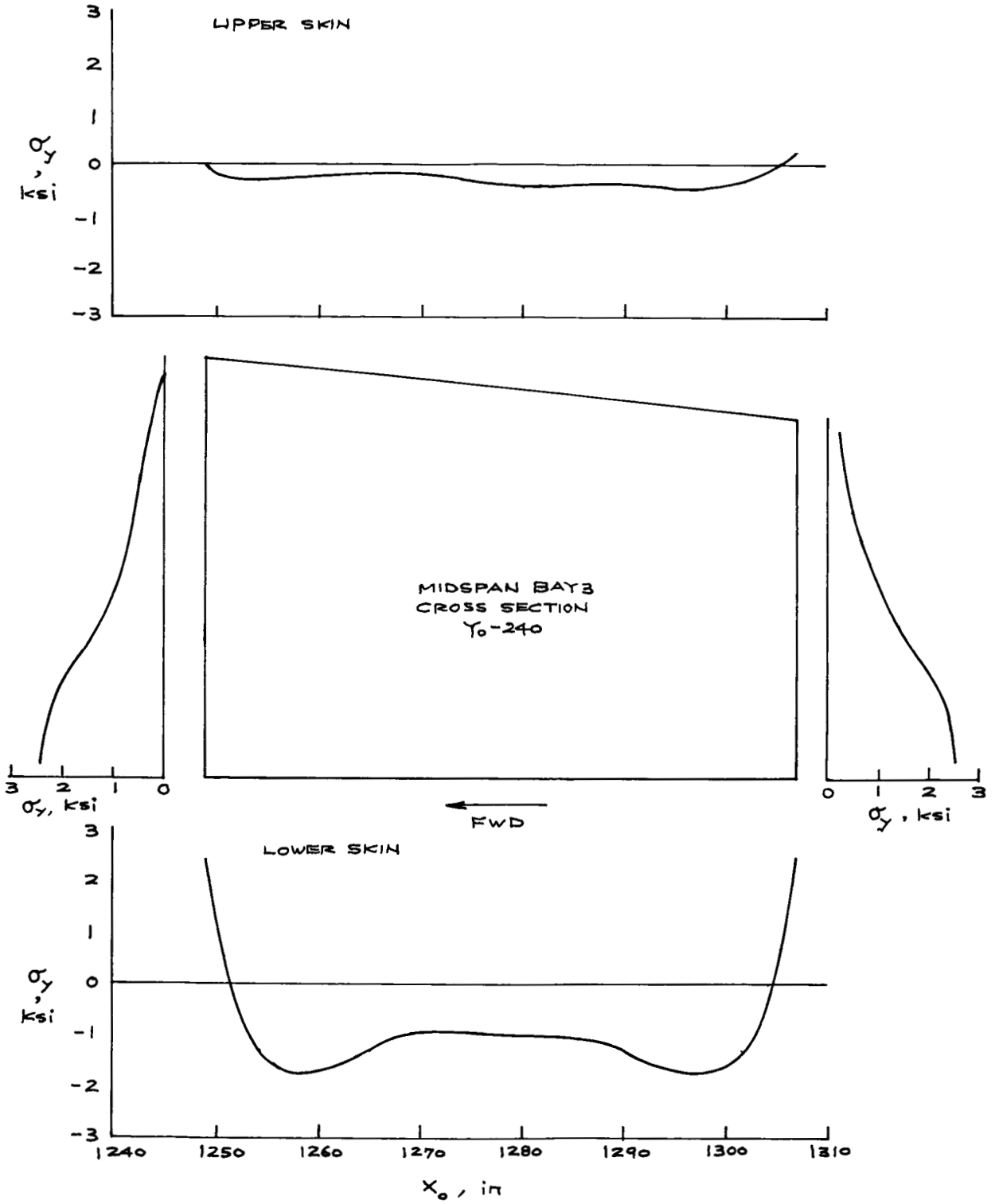


Figure 19. Continuous distributions in the Y_0-240 plane of spanwise stress σ_y based on NASTRAN structural model E; time = 1700 sec, STS-5 thermal loading.

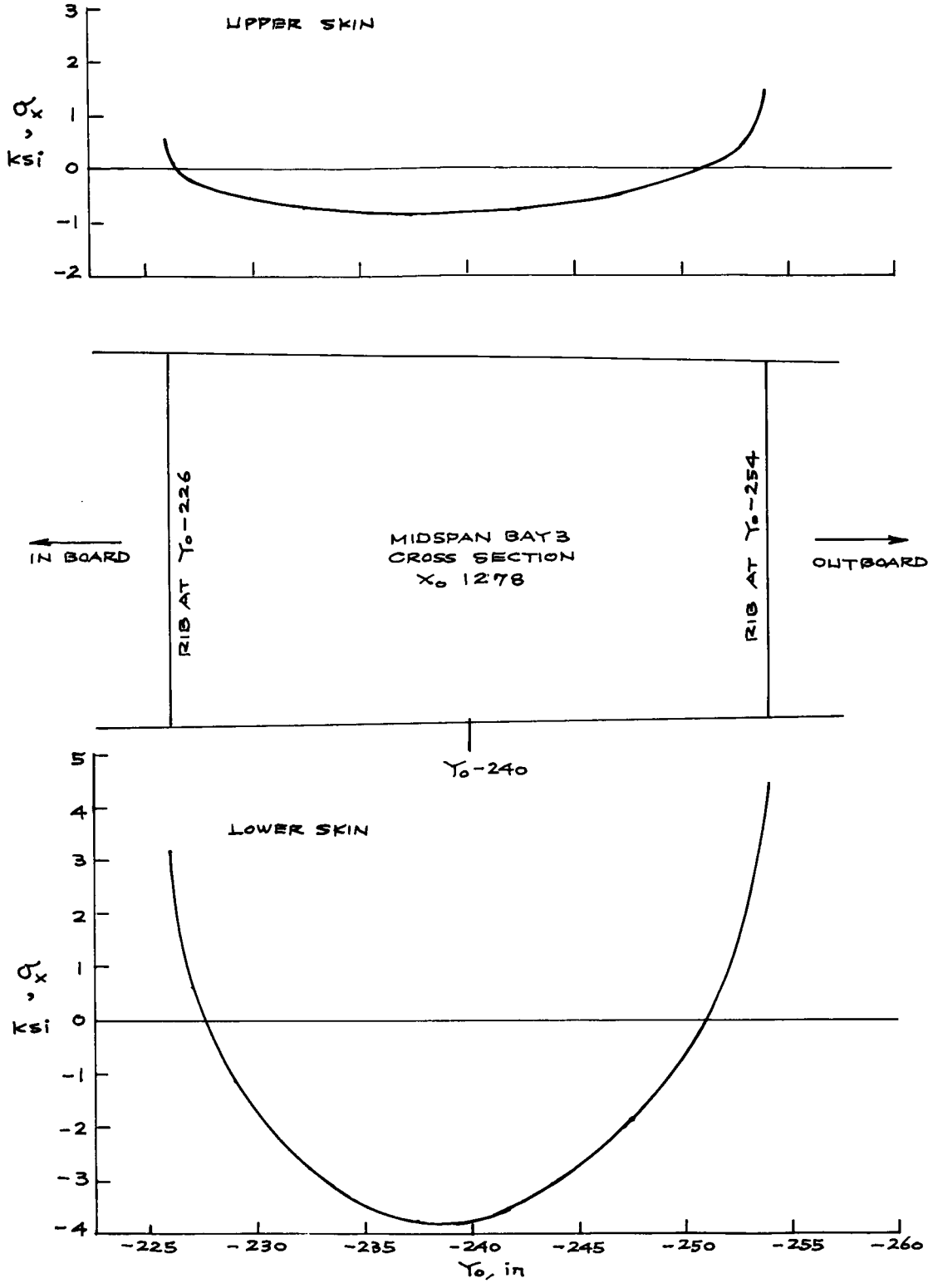


Figure 20. Continuous distributions in the $X_0 1278$ plane of chordwise stress σ_x based on NASTRAN structural model E; time = 1700 sec, STS-5 thermal loading.

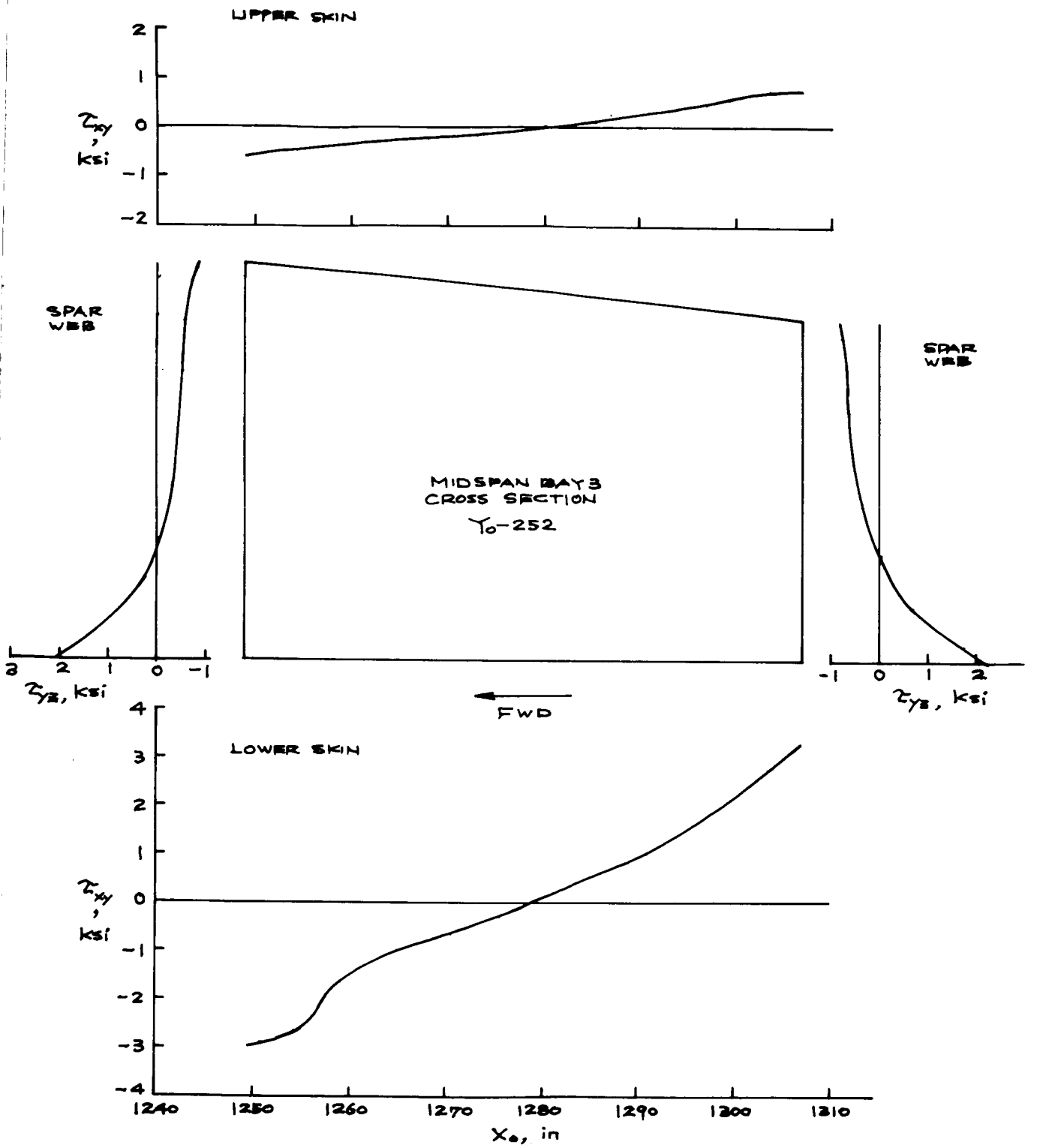


Figure 21. Continuous distributions in the Y_0-252 plane of shear stresses τ_{xy} and τ_{yz} based on NASTRAN structural model E; time = 1700 sec, STS-5 thermal loading.

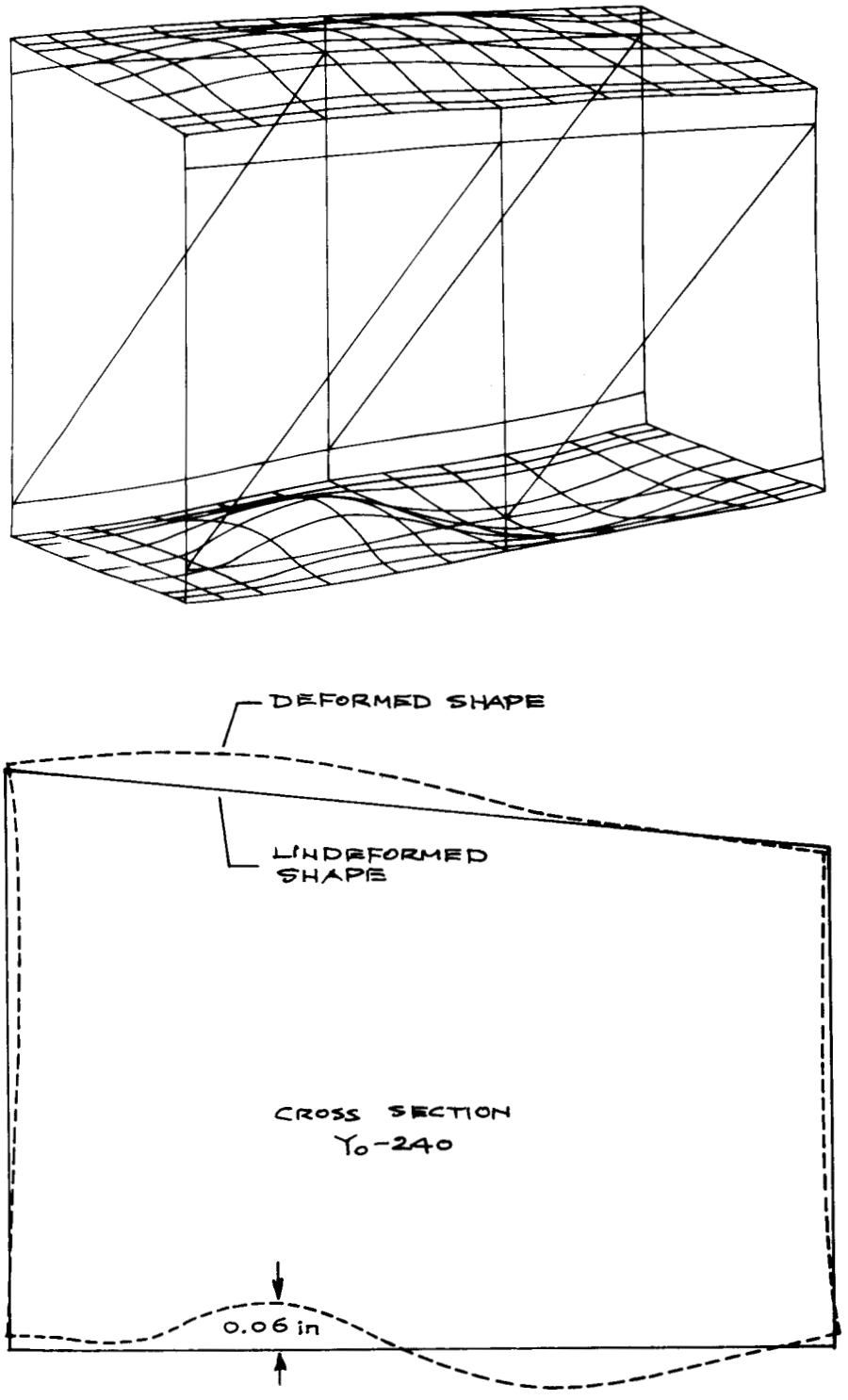


Figure 22. Deformed shape of orbiter wing midspan bay 3 due to STS-5 thermal loading (dimension in inches); time = 1700 sec.

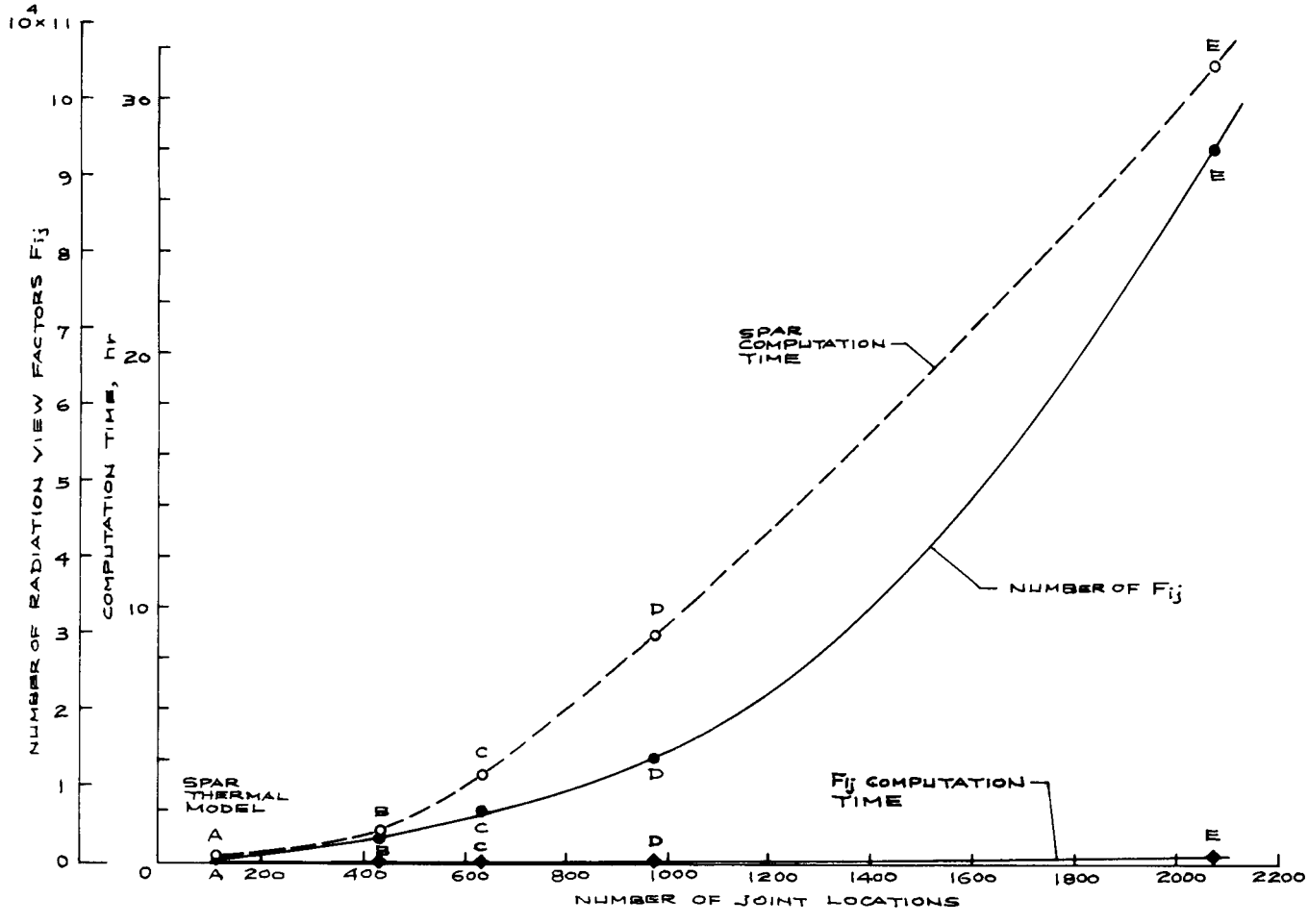


Figure 23. Plots of number of radiation view factors F_{ij} and SPAR computation time as functions of number of joint locations.

STRESS AND VIBRATION ANALYSIS OF RADIAL GAS TURBINE COMPONENTS

Ravi S. Krishnamurthy
Tiernay Turbines, Inc.
Phoenix, AZ 85036

SUMMARY

Predictability of combined deformation of the static structure and the rotor is crucial in determining the build-dimensions for turbomachinery components under the influence of thermal, centrifugal and mechanical loads. Insight into the stress distribution aids in assessing the safety margins as well as the adequacy of containment. Blade natural frequencies that coincide with a multiple of the engine speed need to be avoided. NASTRAN has been used to carry out finite element analysis on critical radial gas turbine parts in conjunction with the Intergraph modeling system in the VAX environment. The CIHEX1 solid element type was used for the stress model whereas the CIHEX2 type was used for the vibration model. The justification for using the higher order element for vibration analysis is explained through experimental verification. Typical results of the analysis together with stress contours and mode shapes obtained are presented.

INTRODUCTION

Finite element analysis has been increasingly used in the design of turbomachinery components (example, ref.1). Advances in graphics modeling systems have facilitated semi-automatic generation of finite element models for more detailed analyses when used together with large analysis programs such as NASTRAN. This paper briefly describes an application of COSMIC/NASTRAN in the analysis of typical radial gas turbine engine parts.

Components with complex shapes such as impellers and turbine wheels necessitate modeling the geometry using 3-D elements, even though the part may exhibit certain symmetry of shape and loading. The choice of the finite element type and size depends upon the complexity of shape as well as the accuracy of analysis desired. Importance of adapting the right type of element for application to turbine blade vibration problem has been previously emphasized (ref. 2,3).

Analysis of rotating components must include centrifugal forces as well as thermal loads in determining the stress distribution and the corresponding mechanical deformation. Thermal and any mechanical loads on the static structure result in independent deformation. Combined deformation under all operating conditions must ensure prevention of rubbing. Geometrical changes aided by analysis at the design stage can assure such a deformation compatibility. In addition, calculation of blade natural frequencies helps in avoiding those frequencies that potentially cause resonance during the engine operation.

In the following presentation, two 3-D COSMIC/NASTRAN element types, CIHEX1 and CIHEX2, have been utilized for stress and vibration analysis of turbine wheels and blades. The reason for using the higher order element type for eigenvalue analysis is explained, and comparisons with theory for a simple model as well as with measured frequency on the actual hardware are included.

MODEL DEVELOPMENT

All models were created from design files interactively on the graphics system. The Intergrapgh system utilized provides a semi-automatic mesh generation capability for the element types used that are compatible with COSMIC/NASTRAN.

Because of the cyclic symmetry features of a rotating component, only a representative section (pie-section) of the part needs to be modeled. However, the complex shape of the blade dictates the use of 3-D solid elements for the pie-section. An example of the finite element model using 8-node hexahedron CIHEX1 solid element is shown in figure 1. The blade portion as well as the two sides of the pie-section of the hub area can be meshed with the nodes faithfully placed on the corresponding B-spline surfaces.

The graphics model is then translated into a NASTRAN-acceptable ASCII data file, with the node coordinates defined in terms of cylindrical coordinate system (CCS) instead or the usual rectangular coordinate system (RCS) for ease of defining proper nodal constraints.

For vibration analysis, the blade model is created using 20-node isoparametric, parabolic solid element, CIHEX2. Appropriate nodes along the blade root are constrained. The element discretization size is largely dependent upon the mesh generation limitations as well as the computational constraints. A higher order element is preferable to a smaller-sized lower-ordered one for evaluating natural frequencies. Figure 2 shows the finite element model of an air-cycle machine inlet fan blade for vibration analysis. COSMIC/NASTRAN also permits the use of a cubic isoparametric element, CIHEX3.

In addition, a simple aluminum cantilever beam of size 50.8 mm x 12.7 mm x 3.175 mm was modeled using both CIHEX1 and CIHEX2 elements. The results for the first mode natural frequency are compared with the theoretical value.

ANALYSIS AND POST-PROCESSING

The stress analysis is carried out in two steps. First, the translated model data deck is appended with thermal boundary conditions data obtained through aerodynamic and thermodynamic analyses. The initial NASTRAN heat transfer analysis run then gives the nodal temperature distribution throughout the model.

Secondly, static analysis is performed with the bulk data deck modified to include the nodal temperatures. The engine rotational speed is include on the RFORCE card. All nodes on the two sides of the pie-section are constrained to eliminate circumferential displacement (components 2456 in CCS contrained on the GRID card). Both the displacement and the element stress outputs are requested on the case control card set. The NASTRAN stress output punch file is edited to extract only the pertinent stress values at the nodes for each element, and then an average stress value is obtained at each node for post-processing convenience.

The displacement and the nodal stress data are loaded into the graphics finite element post-processor in order to display the model deformed shape, and the stress distribution contours. The element center stresses can also be displayed as color-coded elements or color-filled contours after hidden-line removal.

Real eigenvalue analysis using the inverse power method is used for calculating the blade natural frequencies. The 20-node solid element vibration model runs much slower on COSMIC/NASTRAN than the 8-node element model, even when only a few roots are required. However, the CIHEX2 element model yields far more accurate results. The frequency range for eigenvalue search is carefully chosen to include the lowest modes.

A variety of model discretizations were used in order to determine its influence on the accuracy of results. Both CIHEX1 and CIHEX2 models were processed with nodal coordinates translated in single- and double-precision formats. The resulting eigenvectors were loaded into the post-processor for displaying different modes by graphics animation.

RESULTS AND DISCUSSION

The stress contours obtained for the aluminum turbine wheel is shown in figure 3. The value of the maximum stress developed is helpful in determining the available safety margin. The average tangential stress across the hub section is used for calculating the burst speed. The displacement results are used in obtaining build-dimensions in order to assure proper steady-state running clearances.

Figure 4 shows the first two normalized modal vibration patterns analytically obtained. The calculated frequencies for various element configurations are presented in table I, and graphically represented in figure 5 for the first mode frequency. Models #2 and #4 have two-layered elements in the blade thickness direction (2 sectors). All results shown were obtained with double-precision processing. Single-precision processing, although much faster, yields highly erroneous results.

Of particular interest are the trends in accuracy improvement obtained through finer discretization, increased sectors, and the use of higher-order element. Because of the superior behavior of the CIHEX2 element in bending, a relatively coarse finite element grid is adequate in obtaining satisfactory results. This element, however, does not seem to offer additional advantages for static stress calculations. The faster CIHEX1 linear element is generally adequate for that purpose.

The theoretical first mode natural frequency for the cantilever beam modeled is 1,003 Hz. The NASTRAN calculated results, all processed in double-precision, for the two types of solid elements are compared in table II. Different number of elements in the length, width and thickness (L x W x T) directions were used. These results indicate, in a fashion similar to the above comparison with the experimental results, that CIHEX2 performs very well even with relatively few number of elements. Also, further increase in the number of elements does not substantially improve the accuracy of the results.

The NASTRAN results have been compared with frequency mesurement on the actual hardware. The frequencies were obtained while exciting the blade with a magnetic probe, and, separately, using holographic interferometry. However, in an ordinary laboratory situation, the experimental methods generally have the following limitations: it is difficult to include centrifugal stiffening and thermal effects, only lower modes are easy to excite, and may introduce additional stiffening in contact methods.

NASTRAN results for vibration seem to consistently yield a higher value for the lower modes when compared to the actual value. It is well known that centrifugal effects tend to increase the natural frequency of the blade with increasing engine speed. A thorough vibration analysis must include the effect of the centrifugal forces. In this regard, a two-step analysis procedure using a different version of NASTRAN has been reported (ref. 4).

The frequency results are commonly incorporated in a Campbell diagram for the engine - a plot of frequency vs. engine speed - where any interference of a blade resonant frequency with a possible excitation mechanism can be easily isolated. Furthermore, at the design stage, blade frequencies can be tuned by appropriately altering the blade shape (changing the taper ratio, etc.), until the required blade stress and vibration characteristics are found.

REFERENCES

1. Boyd, D.I.: Development of a New Technology Small Fan Jet Engine. ASME 85-IGT-139, International Gas Turbine Symposium and Exposition, Beijing, Peoples Reupublic of China, September, 1985.

2. Rieger, N.F.: Finite Element Analysis of Turbomachine Blade Problems. In: Finite Element Applications in Vibration Problems. ASME Conference, September 1977, Ed.: Kamal, M.M., pp. 93-120.

3. Wachter, J.: Analysis of Impeller Vibrations in Radial Compressors. AMSE 86-GT-219, International Gas Turbine Conference and Exhibit, Dusseldorf West Germany, June 1986.

4. Lawrence, C.; Aiello, R.A.; Ernst, M.A.; and McGee, O.G.: A NASTRAN Primer for the Analysis of Rotating Flexible Blades. NASA Technical Memorandum 89861, May, 1987.

TABLE I. - FAN BLADE CALCULATED NATURAL FREQUENCIES

Element Type	Model#1 CIHEX1	Model#2 CIHEX1+ CWEDGE	Model#3 CIHEX1+	Model#4 CIHEX1+ CWEDGE	Model#5 CIHEX2	Model#6 CIHEX2	Experimental
# Elements (total)	100	180	180	600	90	120	
# Elements (axial)	10	6X2	9	10X2	6	6	
# Elements (radial)	10	15	20	30	15	20	
# Nodes	242	268	286	899	746	973	
	Frequency, Hz						
Mode #1 (bending)	4,123	3,908	3,890	3,538	3,394	3,375	2,950
Mode #2 (twisting)	7,801	8,779	8,618	8,013	7,249	7,264	6,500
Mode #3	13,778	13,697	13,506				
Mode #4	17,222						

TABLE II. - NASTRAN RESULTS FOR THE CANTILEVER BEAM. FIRST MODE NATURAL FREQUENCIES. (THEORETICAL VALUE = 1,003 HZ.)

	CIHEX1 Model			CIHEX2 Model		
	# Elements L x W x T	Frequency	Error	# Elements L x W x T	Frequency	Error
Model # I	5 x 1 x 1	2,285 Hz	128%	5 x 1 x 1	1,047 Hz	4.4%
Model # II	10 x 3 x 2	1,451 Hz	45%	10 x 1 x 1	1,028 Hz	2.5%
Model # III	24 x 3 x 2	1,117 Hz	11%	20 x 1 x 1	1,021 Hz	1.8%

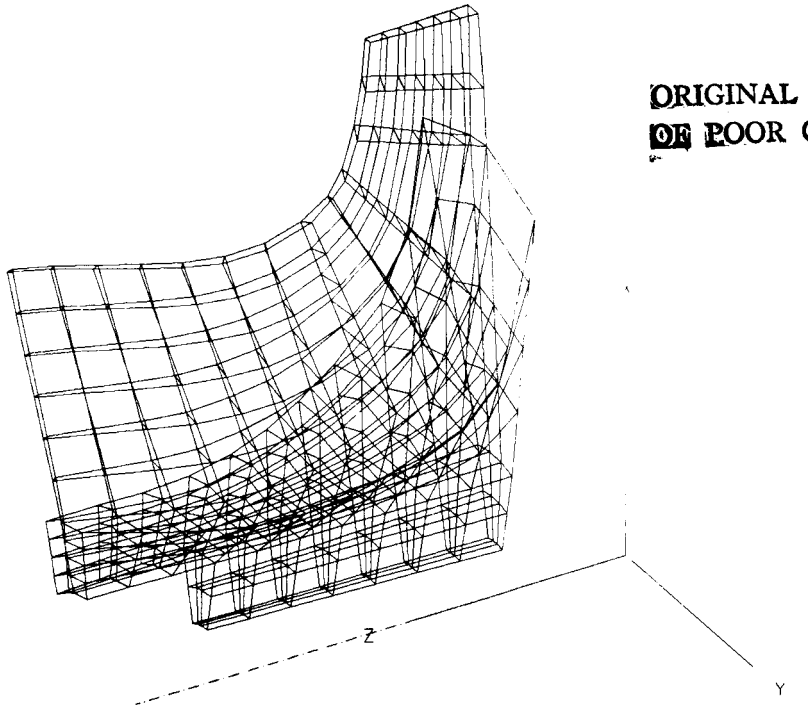


Figure 1 : FE Model for Turbine Wheel using Solid Elements.

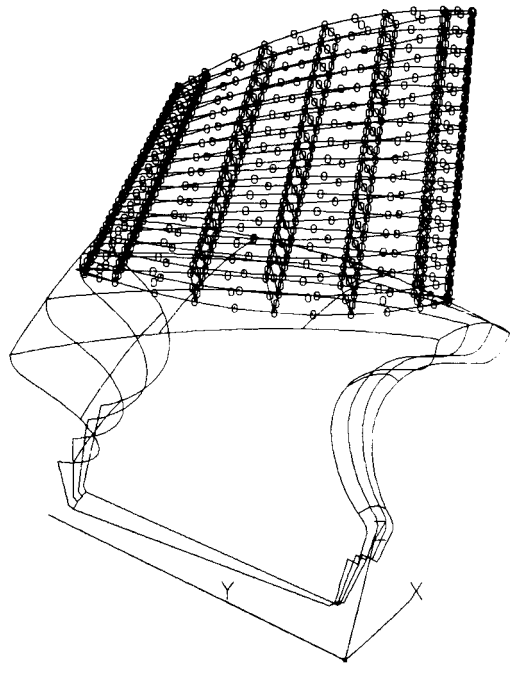


Figure 2 : Fan Blade Model using CIHEX2 Elements.

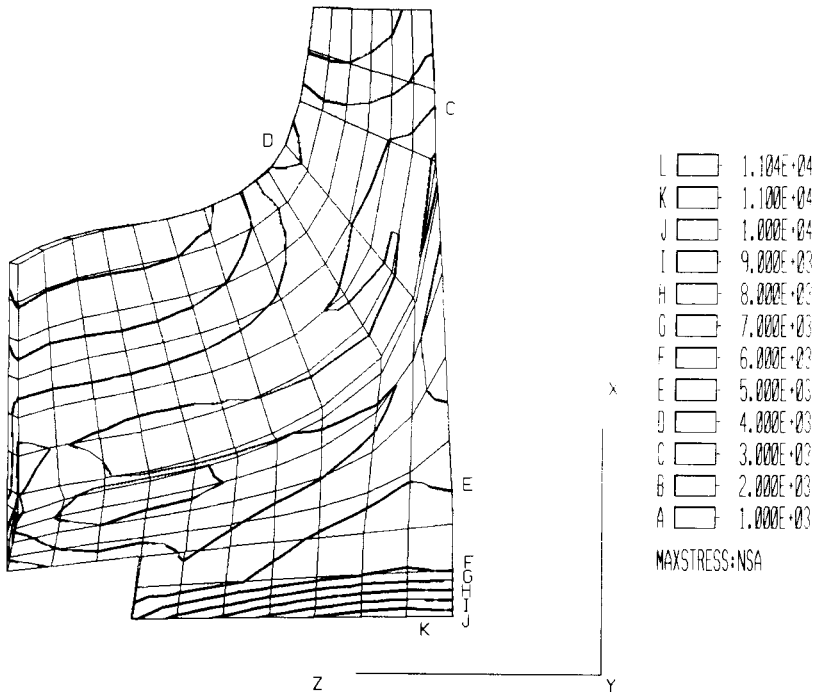
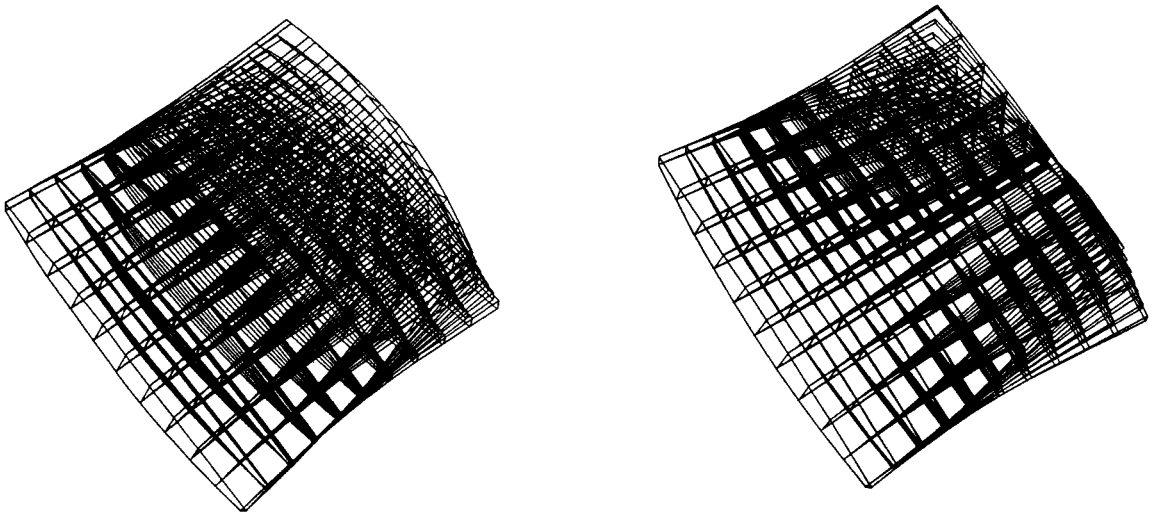


Figure 3 : Stress Contours with Hidden Lines Removed.



Mode #1 Mode #2

Figure 4 : Vibration Patterns for Fan Blade.

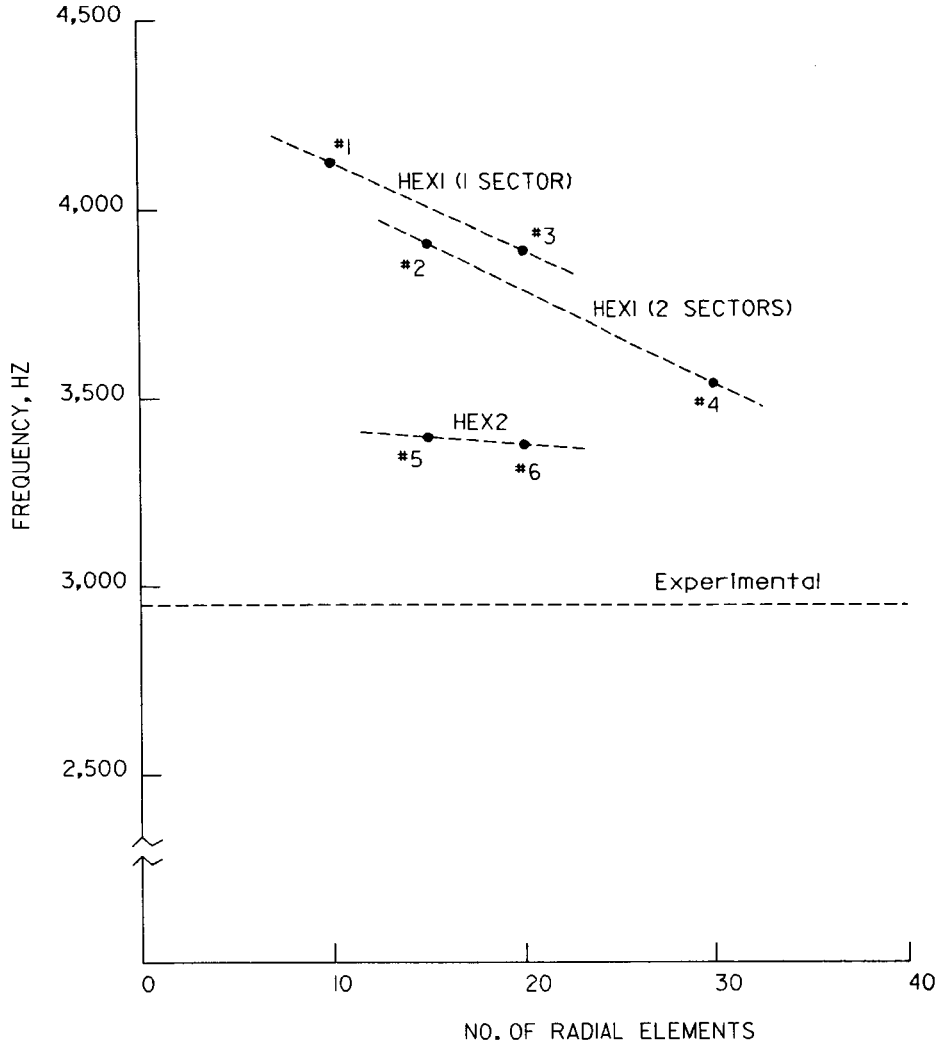


Figure 5 : Blade Natural Frequencies from Alternative Models.

TREATMENT OF STATIC PRELOAD EFFECTS IN ACOUSTIC RADIATION AND SCATTERING

by

Gordon C. Everstine
Applied Mathematics Division (184)
David Taylor Research Center
Bethesda, Maryland 20084 U.S.A.

ABSTRACT

NASHUA is a coupled finite element/boundary element capability built around NASTRAN for calculating the low frequency, far-field acoustic pressure field radiated or scattered by an arbitrary submerged 3-D elastic structure subjected to either internal time-harmonic mechanical loads or external time-harmonic incident loadings. This paper describes the addition to NASHUA of the capability to take into account the effects of static preload on the stiffness of the structure. The static preload is accounted for using NASTRAN's differential stiffness matrix and implemented by merging parts of NASTRAN's differential stiffness rigid formats into the direct frequency response calculation, some of which is done in NASTRAN. The general solution approach calculates structural and fluid impedances with no approximation other than discretization. The surface fluid pressures and normal velocities are first calculated by coupling a NASTRAN finite element model of the structure with a discretized form of the Helmholtz surface integral equation for the exterior fluid. Far-field pressures are then evaluated from the surface solution using an asymptotic form of the Helmholtz exterior integral equation. The effects of adding static preload (e.g., hydrostatic pressure) to the calculation are illustrated for an internally-driven spherical shell.

INTRODUCTION

Two basic problems in numerical structural-acoustics are (1) the calculation of the acoustic pressure field radiated by a general submerged three-dimensional elastic structure subjected to internal time-harmonic loads, and (2) the calculation of the far-field acoustic pressure scattered by an elastic structure subjected to an incident time-harmonic wave train. The most common, as well as the most accurate, general approach for solving these problems is to couple a finite element model of the structure with a boundary element model of the surrounding fluid.[1-5] This is the approach taken by NASHUA, which is a boundary element program built around NASTRAN, a widely-used finite element computer program for structural dynamics.

Two previous papers described the basic development for acoustic radiation and scattering.[4,5] Here we describe the addition to NASHUA of the capability to take into account in the analysis the effects of a static

preload on the stiffness of the structure. The static preload (which may be due, for example, to hydrostatic pressure) is accounted for by using NASTRAN's differential stiffness matrix and is implemented by merging parts of NASTRAN's differential stiffness rigid formats into the direct frequency response calculation, some of which is done in NASTRAN.

In general, the NASHUA procedure uses NASTRAN to generate the structure's stiffness, mass, and damping matrices and to perform various matrix manipulations. Other programs are used to generate the fluid matrices, perform the field calculations, and display the results. The procedure is highly automated, so that a finite element model of a dry structure can often be converted for structural-acoustic analysis with NASHUA in a few hours.

THEORETICAL APPROACH

The basic theoretical development for NASHUA's radiation and scattering approach has been presented in detail previously.[4,5] Here, for completeness, we summarize the overall approach and describe the addition of the hydrostatic preload effects.

The Surface Solution

Consider an arbitrary submerged three-dimensional elastic structure subjected to either internal time-harmonic loads or an external time-harmonic incident pressure wave train. If the structure is modeled with finite elements using NASTRAN, the resulting matrix equation of motion for the structural degrees of freedom (DOF) can be written as

$$Zv = F - GAp, \tag{1}$$

where Z = structural impedance matrix (dimension s x s),

v = complex amplitude of the velocity vector for all structural DOF (wet and dry) in terms of the coordinate systems selected by the user (s x r),

F = complex amplitude of the vector of mechanical forces applied to the structure (s x r),

G = rectangular transformation matrix of direction cosines to transform a vector of outward normal forces at the wet points to a vector of forces at all points in the coordinate systems selected by the user (s x f),

A = diagonal area matrix for the wet surface (f x f), and

p = complex amplitude of total pressures (incident + scattered) applied at the wet grid points (f x r).

In this equation, the time dependence $\exp(i\omega t)$ has been suppressed. In the above dimensions, s denotes the total number of independent structural DOF (wet and dry), f denotes the number of fluid DOF (the number of wet points), and r denotes the number of load cases. In general, surface areas, normals, and the transformation matrix G are obtained in NASHUA from the NASTRAN calculation of the load vector resulting from an outwardly directed static unit pressure load on the structure's wet surface.

In Eq. 1, the structural impedance matrix Z, the matrix which converts velocity to force, is given by

$$Z = (-\omega^2 M + i\omega B + K)/i\omega, \tag{2}$$

where M, B, and K are the structural mass, viscous damping, and stiffness matrices, respectively, and ω is the circular frequency of excitation. For structures with a nonzero loss factor, K is complex. A standard NASTRAN finite element model of the structure supplies the matrices K, M, and B.

The total fluid pressure p satisfies the Helmholtz differential equation

$$\nabla^2 p + k^2 p = 0, \tag{3}$$

where $k = \omega/c$ is the acoustic wave number, and c is the speed of sound in the fluid. Equivalently, p is the solution of the Helmholtz integral equation[2,6]

$$\int_S p(\underline{x})(\partial D(r)/\partial n)dS - \int_S q(\underline{x})D(r)dS = \begin{cases} p(\underline{x}')/2 - p_I, & \underline{x}' \text{ on } S \\ p(\underline{x}'), & \underline{x}' \text{ in } E \end{cases} \tag{4}$$

where S and E denote surface and exterior fluid points, respectively, p_I is the incident free-field pressure, r is the distance from $\underline{x}$ to $\underline{x}'$ (Fig. 1), D is the Green's function

$$D(r) = e^{-ikr}/4\pi r, \tag{5}$$

$$q = \partial p/\partial n = -i\omega\rho v_n, \tag{6}$$

ρ is the mass density of the fluid, and v_n is the outward normal component of velocity on S. As shown in Fig. 1, $\underline{x}$ in Eq. 4 is the position vector for a

typical point P_j on the surface S, $\underline{x}'$ is the position vector for the point P_i which may be either on the surface or in the exterior field E, the vector $\underline{r} = \underline{x}' - \underline{x}$, and $\underline{n}$ is the unit outward normal at P_j. We denote the lengths of the vectors $\underline{x}$, $\underline{x}'$, and $\underline{r}$ by x, x', and r, respectively. The normal derivative of the Green's function D appearing in Eq. 4 can be evaluated as

$$\partial D(r)/\partial n = (e^{-ikr}/4\pi r)\ (ik + 1/r)\ \cos\beta, \tag{7}$$

where β is defined as the angle between the normal $\underline{n}$ and the vector $\underline{r}$, as shown in Fig. 1.

The substitution of Eqs. 6 and 7 into the surface equation (4) yields

$$p(\underline{x}')/2 - \int_S p(\underline{x})\ (e^{-ikr}/4\pi r)\ (ik + 1/r)\ \cos\beta\ dS$$

$$= i\omega\rho \int_S v_n(\underline{x})\ (e^{-ikr}/4\pi r)dS + p_I, \tag{8}$$

where $\underline{x}'$ is on S. This integral equation relates the total pressure p and normal velocity v_n on S. If the integrals in Eq. 8 are discretized for numerical computation (the details of which were presented previously[4]), we obtain the matrix equation

$$Ep = Cv_n + p_I \tag{9}$$

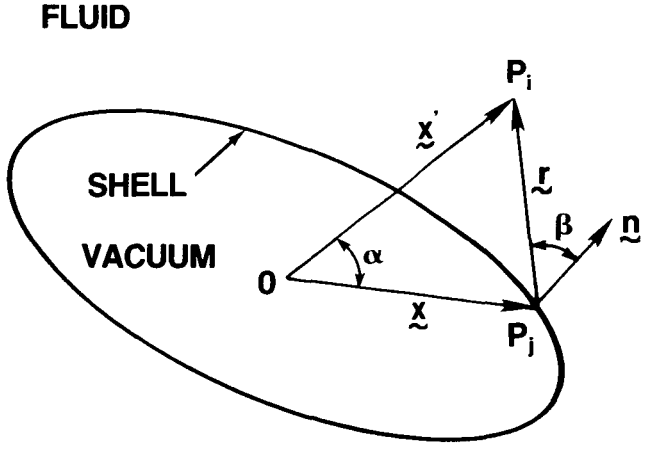


Figure 1 - Notation for Helmholtz Integral Equation

on S, where p is the vector of complex amplitudes of the total pressure on the structure's wet surface, E and C and fully-populated, complex, non-symmetric, frequency-dependent matrices, and p_I is the complex amplitude of the incident pressure vector, if any. The number of unknowns in this system is f, the number of wet points on the fluid-structure interface.

The normal velocities v_n in Eq. 9 are related to the total velocities v by the same rectangular transformation matrix G:

$$v_n = G^T v, \tag{10}$$

where T denotes the matrix transpose. If velocities v and v_n are eliminated from Eqs. 1, 9, and 10, the resulting equation for the coupled fluid-structure system is

$$(E + CG^T Z^{-1} GA)p = CG^T Z^{-1} F + p_I. \tag{11}$$

Since the left-hand side coefficient matrix and the right-hand side of this equation depend on geometry, material properties, and frequency, this equation can be solved to yield the total surface pressures p. Since the two right-hand side terms in Eq. 11 correspond to mechanical and incident loadings, respectively, only one of the two terms would ordinarily be present for a given case. The details of the calculation of the incident pressure vector p_I for scattering problems were presented in an earlier paper[5] and will not be repeated here.

The vector v of velocities at all structural DOF can then be recovered by solving Eq. 1 for v:

$$v = Z^{-1} F - Z^{-1} GAp. \tag{12}$$

Surface normal velocities v_n may be recovered by substituting this solution for v into Eq. 10.

Hydrostatic Pressure Effects

The primary effect of hydrostatic pressure on the dynamics of a submerged structure is to decrease the stiffness of the structure. This decrease, in turn, results in a shift of the resonant frequencies of the shell. NASHUA accounts for this effect by replacing the elastic stiffness matrix K in Eq. 2 with the sum of K and the NASTRAN differential stiffness matrix K_d. Since the user specifies a unit pressure loading on the structure's wet surface (for the purpose of identifying the wet surface and calculating the areas and normals), sufficient information is available to compute K_d, given the desired

hydrostatic pressure. The NASHUA implementation assumes that the pressure is applied uniformly over the wet surface; that is, no depth dependence is accounted for.

If we let P denote the static load vector resulting from the application of the unit outward pressure on the structure's wet surface, the corresponding displacement vector u_s is the solution of

$$K_e u_s = P, \tag{13}$$

where K_e is the real part of the elastic stiffness matrix K. This solution (u_s) is then used by the NASTRAN functional module DSMG1 to compute the differential stiffness matrix K_{do} associated with the unit pressure load. The differential stiffness matrix K_d for the desired hydrostatic pressure p_h is then

$$K_d = - p_h K_{do}, \tag{14}$$

where the minus sign results from the convention that p_h is positive in compression. The final step is the replacement (by equivalencing) of the complex stiffness K by $K + K_d$.

The stiffness matrix K_e is singular for structures which are not sufficiently restrained to prevent rigid body motion, a common occurrence. Since K_e must be nonsingular to solve Eq. 13, the difficulty is resolved by temporarily replacing K_e with the sum of K_e and a diagonal matrix having small positive real numbers on the diagonal. These numbers are 10^{-6} times the corresponding diagonal entries in K_e. This approach relieves the user of having to be concerned with free-body supports for free-free structures. The correction is temporary since it is used only to generate the static solution needed for the differential stiffness calculation and not for the subsequent coupled analysis.

It is important to ensure that the applied hydrostatic pressure p_h is below the lowest buckling load for the structure, since otherwise the differential stiffness matrix K_d would be meaningless. This buckling load can be determined by a separate NASTRAN analysis using Rigid Format 5.

The Far-Field Calculation

With the solution for the total pressures and velocities on the surface, the exterior Helmholtz integral equation, Eq. 4, can be integrated to obtain the radiated (or scattered) pressure at any desired location $\underline{x}'$ in the exterior field. We first substitute Eqs. 6 and 7 into Eq. 4 to obtain a form suitable for numerical integration:

$$p(\underline{x}') = \int_S [i\omega\rho v_n(\underline{x}) + (ik + 1/r)p(\underline{x}) \cos \beta] \, (e^{-ikr}/4\pi r)dS, \qquad (15)$$

where all symbols have the definitions used previously, and $\underline{x}'$ is in the exterior field. Thus, with the total pressure p and normal velocity v_n on the surface S, the radiated or scattered pressure at $\underline{x}'$ can be determined by numerical quadrature using Eq. 15.

In applications, however, the field pressures generally of interest are in the far-field, so we use an asymptotic form[4,7,8] of this equation instead of Eq. 15:

$$p(\underline{x}') = (ike^{-ikx'}/4\pi x') \int_S [\rho c v_n(\underline{x}) + p(\underline{x}) \cos \beta] e^{ikx \cos \alpha} \, dS, \qquad (16)$$

where α is the angle between the vectors $\underline{x}$ and $\underline{x}'$ (Fig. 1), and, for points in the far-field, cos β is computed using

$$\cos \beta \rightarrow \underline{n} \cdot \underline{x}'/x'. \qquad (17)$$

Summary of Theoretical Approach

The NASHUA solution procedure uses NASTRAN to generate the matrices K, M, B, and F and to generate sufficient geometry information so that the matrices E, C, G, A, and p_I can be computed by a separate program called SURF. K includes the differential stiffness effects of the hydrostatic preload, if any. Then, NASTRAN DMAP is used to form the matrices appearing in Eq. 11, which is solved for the total pressures p using the block solver OCSOLVE[9] written by E.A. Schroeder of the David Taylor Research Center especially for this problem. Next, NASTRAN DMAP is used to recover the surface normal velocities v_n and the vector v of velocities at all structural DOF (NASTRAN's "g-set"). This step completes the surface solution. Then, with this solution for the total pressures and velocities on the surface, the asymptotic (far-field) form of the Helmholtz exterior integral equation is integrated in program FAROUT to compute the far-field radiated pressures. Various tables and graphical displays are generated.

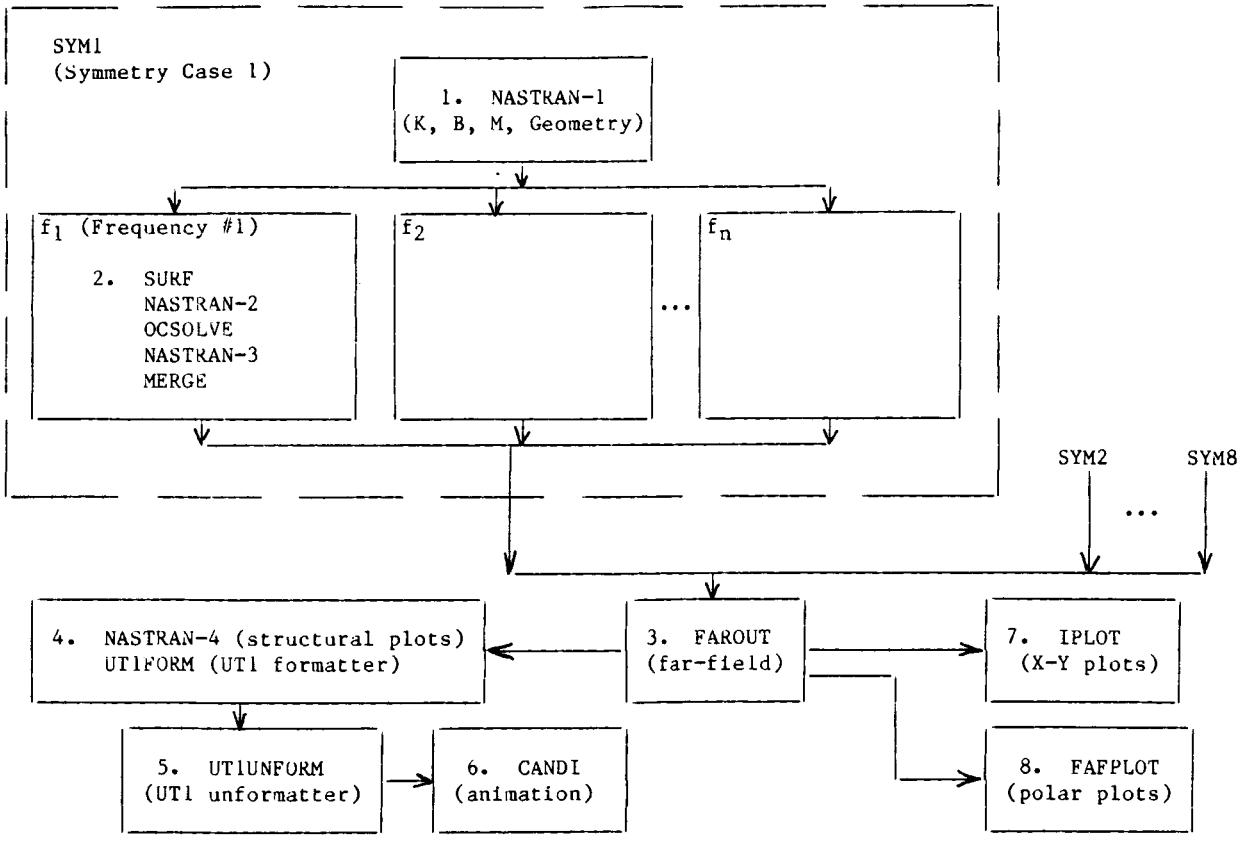
OVERVIEW OF NASHUA SOLUTION PROCEDURE

The overall organization and setup of the solution procedure is summarized in Fig. 2. NASTRAN appears four times in the procedure; to distinguish one NASTRAN execution from another, the integers 1-4 are appended to NASTRAN in the figure.

A separate NASTRAN model is prepared and run (Step 1 in Fig. 2) for each unique set of symmetry constraints. Since up to three planes of reflective

symmetry are allowed, there would be one, two, four, or eight such runs. Step 1 generates files containing geometry information and a checkpoint file for subsequent use in the other steps.

For each symmetry case and drive frequency, the Step 2 sequence is run in a single job. The SURF program reads the geometry file generated by NASTRAN in Step 1 and, using the Helmholtz surface integral equation, generates the fluid matrices E and C for the exterior fluid, the area matrix A, the structure-fluid transformation matrix G, the incident pressure vector p_I, and a geometry file to be used later by FAROUT in Step 3 for the field calculation. SURF is followed by a NASTRAN job which takes the matrices K, M, B, and F from Step 1 and the matrices E, C, A, G, and p_I from SURF and forms the matrices in Eq. 11, which is solved for the total surface pressure vector p by program OCSOLVE.[9] The OCSOLVE program is a general block solver for large, full, complex, nonsymmetric systems of linear, algebraic equations. The program was designed to be particularly effective on such systems and executes on CDC computers about 20 times faster than NASTRAN's equation solver, which was not designed for efficient solution of such systems of



NOTE: Each solid block is a separate job submission.

Figure 2 - Summary of NASHUA Solution Procedure

equations. NASTRAN is then re-entered in Step 2 with p so that the velocities v and v_n can be recovered using DMAP operations. The surface pressures, normal velocities, and full g-set displacements are then reformatted, sorted, and merged into a single file (for each symmetry case) using program MERGE. Recall that there are one, two, four, or eight possible symmetry cases.

Steps 1 and 2 are repeated for each symmetry case. After all symmetry cases have been completed and merged, program FAROUT (Step 3) is run to combine the symmetry cases and to integrate over the surface. FAROUT uses as input the geometry file generated by SURF (Step 2) and the surface solutions from the one, two, four, or eight files generated by MERGE (Step 2). The far-field pressure solution is obtained by integrating the surface pressures and velocities using the asymptotic (far-field) form of the exterior Helmholtz integral equation, Eq. 16. Output from FAROUT consists of both tables and files suitable for various types of plotting.

The remaining steps in the NASHUA procedure are for graphical display. Deformed structural plots of the frequency response are obtained by restarting NASTRAN (Step 4) with the checkpoint file from Step 1 and a results file from FAROUT. In addition, animated plots can be generated on the Evans & Sutherland PS-330 graphics terminal using the CANDI program (Step 6) written for the DEC/VAX computer by R.R. Lipman of DTRC.[10] If the rest of NASHUA is run on a computer other than the VAX, the NASTRAN UT1 file passed to CANDI must first be formatted (Step 4) for transfer to the VAX computer and then unformatted (Step 5) for reading by CANDI.

X-Y plots of various quantities (both surface and far-field) versus frequency may be obtained using the general purpose interactive plotting program IPLOT[11] (Step 7). Polar plots of the far-field sound pressure levels in each of the three principal coordinate planes can also be generated using the interactive graphics program FAFPLOT[12] (Step 8) written by R.R. Lipman.

DMAP ALTER

Several DMAP alters are used in the overall NASHUA procedure. However, the only alter affected by a static preload is that of Step 1, which makes available to NASHUA several geometry data blocks and computes the structural matrices K, M, and B. The hydrostatic pressure option is invoked with the addition of only one bulk data card, a parameter card on which the new parameter HSP (the hydrostatic pressure) is defined. In general, the complete alter for NASTRAN's direct frequency response rigid format now involves two modifications, the generation of the static load vector resulting from the application of the unit pressure load and the calculation of the differential stiffness matrix K_d so that the elastic stiffness matrix K can be replaced by the sum of K and K_d. For the 1987 release of NASTRAN, the following alter is used:

```
ALTER 1 $ NASHUA STEP 1, COSMIC 1987 RF8 (REVISED 12/14/87)
ALTER    2,2 $ DELETE PRECHK
ALTER    21,21 $ REPLACE GP3
```

```
GP3       GEOM3,EQEXIN,GEOM2/SLT,GPTT/S,N,NOGRAV/NEVER=1 $ SLT
ALTER     117,117 $ REPLACE FRRD
SSG1      SLT,BGPDT,CSTM,SIL,EST,MPT,GPTT,EDT,MGG,CASECC,DIT/
          PG/LUSET/NSKIP $ PG
SSG2      USET,GM,YS,KFS,GO,DM,PG/QR,PO,PS,PL $ PL
OUTPUT2   BGPDT,EQEXIN,USET,PG,PL $
OUTPUT2   CSTM,ECT,,, $
OUTPUT2   ,,,, //-9 $
PARAMR    //*EQ*//C,Y,HSP=0./0.////NOHSP $
COND      LBL4D,NOHSP $ SKIP DIFF. STIFF. IF NO HYDROSTATIC PRESSURE
PARAMR    //*COMPLEX//C,Y,HSP=0./0./HSPC $ HSP+I*0
DIAGONAL  KAA/KDIAG/*SQUARE*/1.0 $
ADD       KAA,KDIAG/KAAD//(1.E-6,0.) $
RBMG2     KAAD/LLL $ FACTOR KAA
SSG3      LLL,KAAD,PL,LOO,KOO,PO/ULV,UOOV,RULV,RUOV/OMIT/V,Y,IRES=-1/
          1/S,N,EPSI $ STATIC SOLUTION
SDR1      USET,PG,ULV,UOOV,YS,GO,GM,PS,KFS,KSS,/UGV,PGG,QG/1/
          *BKLO* $ RECOVER DEPENDENT DISPLACEMENTS
TA1       ECT,EPT,BGPDT,SIL,GPTT,CSTM/X1,X2,X3,ECPT,GPCT/LUSET/
          NOSIMP/0/NOGENL/GENEL $ TABLES FOR DIFF. STIFFNESS
DSMG1     CASECC,GPTT,SIL,EDT,UGV,CSTM,MPT,ECPT,GPCT,DIT/KDGG/
          S,N,DSCOSET $ DIFF. STIFF. MATRIX
EQUIV     KDGG,KDNN/MPCF2/MGG,MNN/MPCF2 $ EQUIV IF NO MPC'S
COND      LBL1D,MPCF2 $ TRANSFER IF NO MPC'S
MCE2      USET,GM,KDGG,,,/KDNN,,, $ MPC'S ON DIFF. STIFF.
LABEL     LBL1D $
EQUIV     KDNN,KDFF/SINGLE/MNN,MFF/SINGLE/ $ EQUIV. IF NO SPC'S
COND      LBL2D,SINGLE $ TRANSFER IF NO SPC'S
SCE1      USET,KDNN,,,/KDFF,KDFS,KDSS,,, $ SPC'S AND DIFF. STIFF.
LABEL     LBL2D $
EQUIV     KDFF,KDAA/OMIT/MFF,MAA/OMIT $ EQUIV. IF NO OMITS
COND      LBL3D,OMIT $ TRANSFER IF NO OMITS
SMP2      USET,GO,KDFF/KDAA $ OMITS AND DIFF. STIFF.
LABEL     LBL3D $
PARAMR    //*SUBC*////MHSPC//HSPC $ NEGATE HYDROSTATIC PRESSURE
ADD       KDD,KDAA/NEWKDD//MHSPC $ ADD ELASTIC K AND DIFF. STIFF.
ADD       KFS,KDFS/NEWKFS//MHSPC $ ADD ELASTIC K AND DIFF. STIFF.
EQUIV     NEWKDD,KDD//NEWKFS,KFS $
LABEL     LBL4D $ END OF DIFF. STIFF. EFFECTS (HSP)
DIAGONAL  KDD/IDENT/*SQUARE*/0. $ D-SET IDENTITY
ADD       IDENT,/IDM $ ANOTHER D-SET IDENTITY
ADD       IDENT,/ZERO/(0.0,0.0) $ D-SET ZERO MATRIX
FRRD      CASEXX,USETD,DLT,FRL,GMD,GOD,IDENT,ZERO,IDM,,DIT/
          UDVF,PSF,PDF,PPF/*DISP*/*DIRECT*/LUSETD/MPCF1/
          SINGLE/OMIT/NONCUP/FRQSET $ PDF, KDD=MDD=I, BDD=0
CHKPNT    MDD,KDD,BDD,PDF,PSF,PPF,EQDYN,USETD,GOD,GMD $
CHKPNT    KFS,BGPDT,ECT,EQEXIN,GPECT,SIL $
EXIT      $
ENDALTER $
```

EXAMPLE

Here we illustrate the effect of a hydrostatic pressure preload on the dynamics of a submerged structure by solving the acoustic radiation problem of a submerged thin spherical shell with a distributed internal driving force, as shown in Fig. 3. The particular problem solved has a uniform internal pressure load applied over the polar angle γ = 36 degrees.

We solve with NASHUA the problem with the following characteristics:[13]

$a = 5$ m	shell radius
$h = 0.15$ m	shell thickness
$E = 2.07 \times 10^{11}$ Pa	Young's modulus
$\nu = 0.3$	Poisson's ratio
$\rho_s = 7669$ kg/m^3	shell density
$\eta = 0$	shell loss factor
$\rho = 1000$ kg/m^3	fluid density
$c = 1524$ m/s	fluid speed of sound
$p_o = 1$ Pa	internal pressure
$\gamma = 36°$	extent of internal pressure
$p_h = 1 \times 10^8$ Pa	hydrostatic pressure

The same shell was used previously[4,5] for the validation of the basic radiation and scattering capability in NASHUA. One octant of the shell was modeled with NASTRAN's CTRIA2 membrane/bending elements as shown in Fig. 4. With 20 elements along each edge of the domain, the model has 231 wet points and 1263 structural DOF. Three planes of symmetry were imposed. The application of NASTRAN's buckling analysis (Rigid Format 5) to this shell

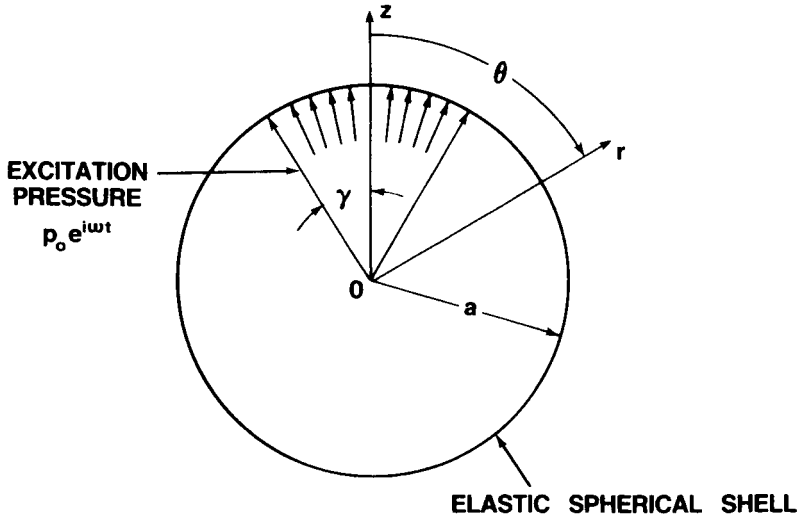


Figure 3 - Submerged Elastic Spherical Shell Driven over Sector

showed that the hydrostatic pressure preload p_h is about 41% of the lowest buckling load of 2.42×10^8 Pa.

The NASHUA model was run for 19 drive frequencies in the nondimensional frequency range ka = 1.0 to ka = 2.05, where a is the shell radius. This frequency range was selected because it includes the first two submerged resonances of the shell (at ka = 1.606 and ka = 1.999) and is below all the discrete critical frequencies at which the surface Helmholtz integral equation (4) is invalid.[14,15] In Fig. 5 we compare the far-field radiated pressure on the polar axis as computed by NASHUA (including the effects of hydrostatic preload) with a converged series solution as computed by Henderson's RADSPHERE program,[16] which assumes zero preload. (Since it was shown previously[4,5] that, for zero preload, NASHUA and RADSPHERE yielded essentially identical results for this problem, it was more economical to use RADSPHERE, rather than NASHUA, to generate the unpressurized solution. RADSPHERE was developed from equations published in the Junger and Feit book.[17]) The ordinate in Fig. 5 is the normalized pressure $|p_r r/p_o a|$, where p_r is the far-field pressure radiated outward along the polar axis at distance r from the origin, and p_o is the magnitude of the internal pressure applied internally over the sector. Clearly, the effect of the hydrostatic preload is to lower slightly the frequencies of the resonances.

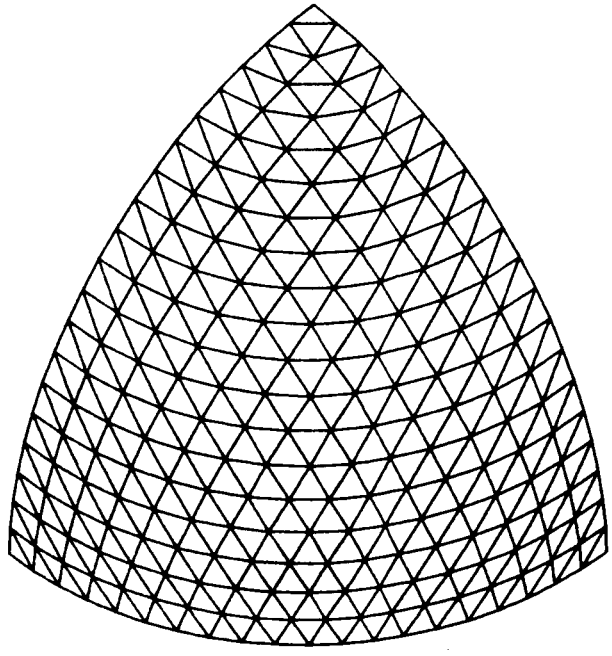


Figure 4 - Finite Element Model of One Octant of Spherical Shell

DISCUSSION

NASHUA is a very general capability built around NASTRAN for predicting the acoustic sound pressure field radiated or scattered by arbitrary three-dimensional elastic structures subjected to time-harmonic loads. Sufficient automation is provided so that, for many structures of practical interest, an existing NASTRAN structural model can be adapted for NASHUA acoustic analysis within a few hours.

One of the major benefits of having NASHUA linked with NASTRAN is the ability to integrate the acoustic analysis of a structure with other dynamic analyses. Thus the same finite element model can be used for modal analysis, frequency response analysis, linear shock analysis, and underwater acoustic analysis. In addition, many of the pre- and postprocessors developed for use with NASTRAN become available for NASHUA as well.

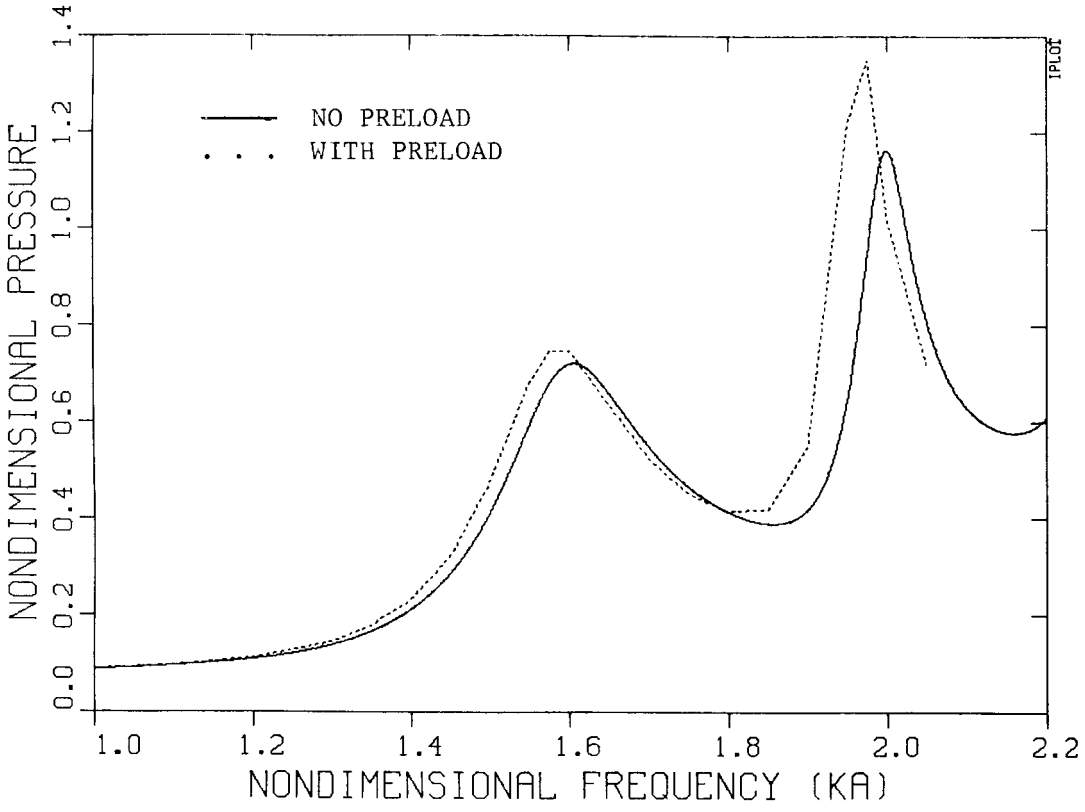


Figure 5 - Normalized Far-Field Pressure $|p_r r/p_o a|$ Radiated Outward Along the Polar Axis with and without a Hydrostatic Preload; Solid Curve Is Solution without Preload, and Dotted Curve Is Solution with Preload.

REFERENCES

1. Chen, L.H., and D.G. Schweikert, "Sound Radiation from an Arbitrary Body," J. Acoust. Soc. Amer., Vol. 35, No. 10, pp. 1626-1632 (1963).

2. Wilton, D.T., "Acoustic Radiation and Scattering From Elastic Structures," Int. J. Num. Meth. in Engrg., Vol. 13, pp. 123-138 (1978).

3. Mathews, I.C., "A Symmetric Boundary Integral-Finite Element Approach for 3-D Fluid Structure Interaction," in Advances in Fluid-Structure Interaction - 1984, PVP-Vol. 78 and AMD-Vol. 64, ed. by G.C. Everstine and M.K. Au-Yang, American Society of Mechanical Engineers, New York, pp. 39-48 (1984).

4. Everstine, G.C., F.M. Henderson, E.A. Schroeder, and R.R. Lipman, "A General Low Frequency Acoustic Radiation Capability for NASTRAN," NASA CP-2419, Fourteenth NASTRAN Users' Colloquium, National Aeronautics and Space Administration, Washington, DC, pp. 293-310 (May 1986).

5. Everstine, G.C., F.M. Henderson, and L.S. Schuetz, "Coupled NASTRAN/ Boundary Element Formulation for Acoustic Scattering," NASA CP-2481, Fifteenth NASTRAN Users' Colloquium, National Aeronautics and Space Administration, Washington, DC, pp. 250-265 (May 1987).

6. Lamb, H., Hydrodynamics, sixth edition, Dover Publications, New York (1945).

7. Chertock, G., "Integral Equation Methods in Sound Radiation and Scattering from Arbitrary Surfaces," NSRDC Report 3538 (1971).

8. Henderson, F.M., "A Structure-Fluid Interaction Capability for the NASA Structural Analysis (NASTRAN) Computer Program," NSRDC Report 3962 (1972).

9. Schroeder, E.A., "A New Block Solver for Large, Full, Unsymmetric, Complex Systems of Linear Algebraic Equations," DTRC-88/003, David Taylor Research Center, Bethesda, Maryland (Jan 1988).

10. Lipman, R.R., "Computer Animation of Modal and Transient Vibrations," Fifteenth NASTRAN Users' Colloquium, NASA CP-2481, National Aeronautics and Space Administration, Washington, DC, pp. 88-97 (May 1987).

11. Everstine, G.C., "A Portable Interactive Plotter for Digital X-Y Data," Report CMLD-86-45, David Taylor Naval Ship R&D Center, Bethesda, Maryland (Dec 1986).

12. Lipman, R.R., "Calculating Far-Field Radiated Sound Pressure Levels from NASTRAN Output," Fourteenth NASTRAN Users' Colloquium, NASA CP-2419, National Aeronautics and Space Administration, Washington, DC, pp. 282-292 (May 1986).

13. Huang, H., and Y.F. Wang, "Asymptotic Fluid-Structure Interaction Theories for Acoustic Radiation Prediction," J. Acoust. Soc. Amer., Vol. 77, No. 4, pp. 1389-1394 (1985).

14. Schenck, H.A., "Improved Integral Formulation for Acoustic Radiation Problems," J. Acoust. Soc. Amer., Vol. 44, No. 1, pp. 41-58 (1968).

15. Huang, H., "Helmholtz Integral Equations for Fluid-Structure Interaction," Advances in Fluid-Structure Interaction - 1984, AMD-Vol. 64, ed. by G.C. Everstine and M.K. Au-Yang, American Society of Mechanical Engineers, New York (1984).

16. Henderson, F.M., "RADSPHERE -- A Computer Program for Calculating the Steady-State, Axially Symmetric, Forced Response and Radiation Field of a Submerged Spherical Shell," DTNSRDC-87/031, David Taylor Naval Ship R&D Center, Bethesda, MD (Aug 1987).

17. Junger, M.C., and D. Feit, Sound, Structures, and Their Interaction, second edition, The MIT Press, Cambridge, Massachusetts (1986).

A MAGNETOSTATIC NONLINEAR MODEL OF A ROTATING ARMATURE PRINTHEAD

T. J. SHEERER
TEXAS INSTRUMENTS INCORPORATED
PERIPHERAL PRODUCTS DIVISION
DATA SYSTEMS GROUP
TEMPLE, TEXAS

1 ABSTRACT:

Using the COSMIC NASTRAN computer program with modifications to allow modelling of nonlinear magnetic materials, a model was made of a rotating armature printhead such as is used in the majority of dot matrix computer printers. The results from the model were compared with experimental data and with the results of an approximate calculation and found to be in good agreement. Using a modification to NASTRAN which output element magnetic quantities directly to the PATRAN pre- and postprocessor, plots of magnetic flux, permeability and field energy were produced which provided a useful illustration of the parameters controlling the device's performance.

2 DESCRIPTION OF A ROTATING ARMATURE PRINTHEAD:

A rotating armature printhead consists of several actuator assemblies such as are shown in Fig.(1) and Fig.(2). Each assembly comprises a magnetic core (A) and actuation coil (B), with a magnetic iron armature (C) pivoted on one leg of the core. The armature is in contact with a print wire (D) and is held in an equilibrium position by a return spring (E) such that a working air gap (F) exists between the armature and one leg of the core. Application of electrical current to the coil results in a magnetic traction force tending to close the working gap, thus accelerating the wire toward the print media (G).

Fig.(3) shows the magnetization curve, B vs. H, of the material used for both the core and the armature, 3 percent Silicon Iron. To maximize the efficiency of the device it is necessary to operate at as high a magnetization level as possible, while keeping the permeability, $\mu = B/H$ large. An accurate analysis of the circuit will allow use of a current level just short of saturating the armature or core. Fig.(4) shows the permeability and also the differential permeability dB/dH of the material.

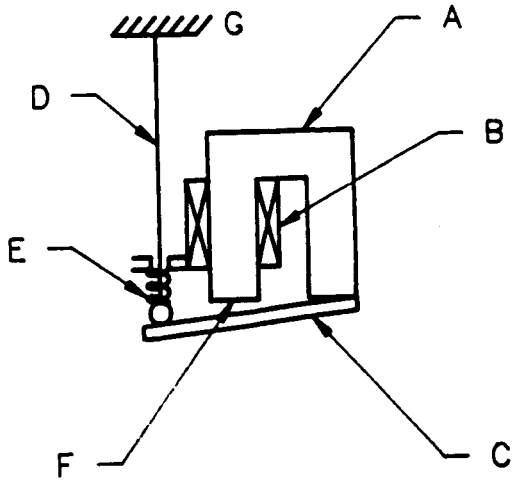


Fig.(1): Rotating Armature Printhead Type 1

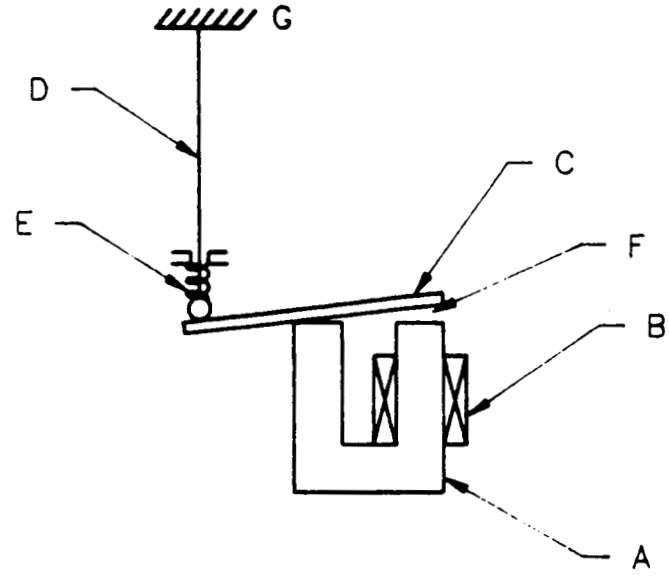


Fig.(2): Rotating Armature Printhead Type 2

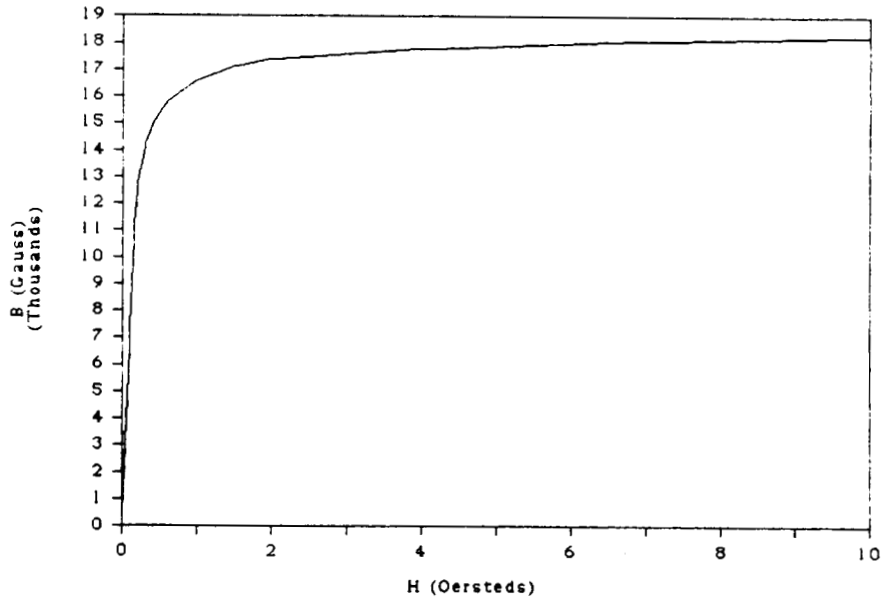


Fig.(3): Magnetization Curve of Silicon Iron

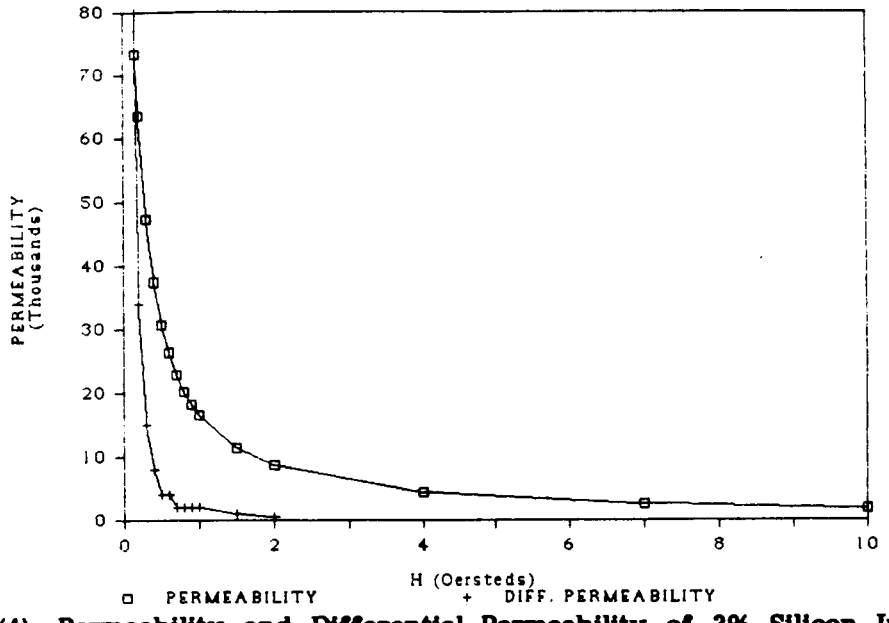


Fig.(4): Permeability and Differential Permeability of 3% Silicon Iron

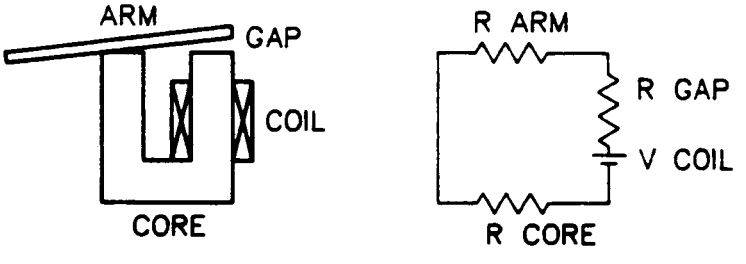


Fig.(5): Electrical Analog of Magnetic Circuit

3 CLASSICAL SOLUTION OF THE BASIC EQUATIONS OF MAGNETOSTATICS:

The basic quantities and equations of magnetostatics are listed in Table (1). It is customary to picture the circuit as the analog of an electrical network in which the source of MMF is equivalent to a voltage source and the various components of the magnetic circuit are analogous to electrical resistors. A schematic of the electrical analog of the magnetic circuit is shown in Fig.(5). The magnetomotive force (MMF) of the coil is readily obtained for any given current level, and a solution obtained by equating the magnetic potential drop around the circuit to the MMF and combining this with the condition of continuity of flux around the circuit to obtain the level of magnetic flux in the circuit:

$$\Phi = \Sigma_n H_n \cdot l_n \tag{1}$$

$$\phi = B_n \cdot \sigma_n \tag{2}$$

$$B = \mu \cdot H \tag{3}$$

This approach neglects the fact that some flux passes through the air surrounding the assembly, and also treats the source of flux as a discontinuity in the Magnetic Potential Φ at the center of the coil. It is also difficult (although not impossible) to allow for variation in permeability due to nonlinear material properties. Fig.(6) shows the magnetic potential and magnetic field in a close path around the circuit in actuality and using this approximation.

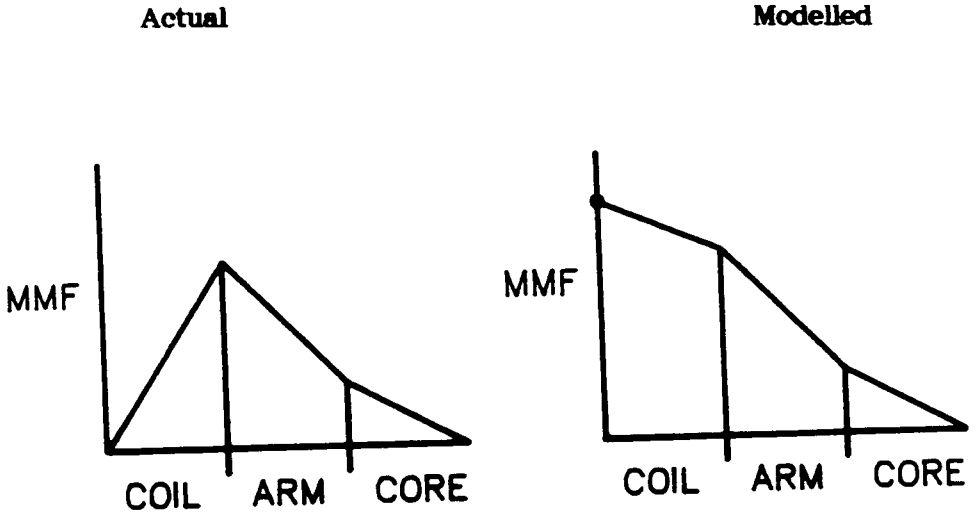


Fig.(6): Magnetic Potential And Field around the Magnetic Circuit

TABLE 1: MAGNETOSTATIC QUANTITIES

QUANTITY	SYMBOL	UNITS	EQUATION
MAGNETIC POTENTIAL	V	GILBERT	$V = \int H \cdot ds$
MAGNETIC FIELD	H	OERSTED	$H = -\nabla V$
MAGNETIC INDUCTION	B	GAUSS	$B = \mu \cdot H$
MAGNETIC FLUX	ϕ	MAXWELL	$\phi = B \cdot \sigma$
RELATIVE PERMEABILITY	μ		$\mu = B/H$
MAGNETOMOTIVE FORCE	Φ	GILBERT	$\Phi = N \cdot i$
ENERGY DENSITY	U	GAUSS-OERSTED	$U = 1/2 \cdot H \cdot B$

4 MODIFICATIONS TO COSMIC NASTRAN TO ALLOW MODELLING OF NONLINEAR MAGNETIC MATERIALS:

NASTRAN already has the capability to model linear magnetic circuits, including the magnetic fields due to arbitrary configurations of current sources, using its existing heat transfer capabilities (1). The use of this capability is described in the NASTRAN user's manual (2). The modification to allow modelling of nonlinear magnetic permeability involved the addition of modules which, after a linear heat transfer solution is obtained and the potential gradient calculated for all elements, examines a table describing the magnetization curve of the material, such as in Fig.(3), and assigns a new value to the element permeability. The element data in the element stiffness table HKELM are multiplied by the ratio of old to new permeability values and a new global stiffness table HKGG is formed from the updated HKELM. The heat transfer solution is now repeated for as many iterations as are required to obtain a converged solution. The DMAP ALTER statements and additional subprograms required have been described elsewhere (3).

There are several ways to interpret the magnetic field, H, all of which are equally valid. In the class of problems where an external source such as a current loop or the earth's magnetic field provides the MMF, the component of H due to the source may be analytically calculated from the Biot-Savart equation:

$$dH_s = i \cdot dl \times \vec{r}/r^3 \qquad (4)$$

where H_s is the field due to the current element dl at distance r.

In this approach NASTRAN is used to calculate the component of H due to the presence of ferromagnetic materials, H_m using

$$H_m = -\nabla \Phi_m \qquad (5)$$

This method is known as the reduced scalar potential method (RSP), and Hurwitz refers to H_m as the anomaly field in (2). The total field is obtainable by summation of the two components. This approach has the great advantage of allowing the modelling of arbitrary current sources, which need not be located within the finite element model. It is described in considerable theoretical detail by Simkin and Trowbridge in (4) and McDaniel et al. in (5). A potential source of error lies in the near-cancellation of H_s and H_m in regions of high permeability, for models containing air gaps. The vector summation of these components is a small difference between two large numbers, which is a classic receipt for arithmetic error. Although such errors are reported in (4), using a special purpose program, the results in (5), using MSC/NASTRAN were accurate for the test cases examined, and COSMIC/NASTRAN is expected to produce equally accurate results. This method requires, however that the source of H_s be decoupled from the effects of the anomaly field, which is not the case if a permanent magnet is modelled, as the magnet is a nonlinear source of MMF which is affected by the magnetic properties of its environs. While it is possible to model such a magnet as a combination of coil and nonlinear material, as is, done, for example, in the AOS/MAGNETIC program, it has been preferred here to use the total scalar potential approach historically used in permanent magnet work, wherein the source of MMF is considered to have a negative permeability and B is opposite in sense to H. In this approach the field is related to potential simply by:

$$H = -\nabla \phi \qquad (6)$$

Just as a coil combined with a magnetic material can simulate a magnet in the method described above, in the total scalar potential method the case of a coil may be handled by replacing the coil with a magnet having appropriate material properties. If the coil modelled is around a high permeability member, accurate results are also obtainable by introducing a discontinuity in potential at the coil center using a value obtained from equation (1). While this is not as elegant as using negative permeability values for the materials enclosed by the coil, it has the advantage of requiring less computer time. While models having positive values of permeability require a 5 or 10 percent damping factor in their iterative solution, a value of 90 percent has been found adviseable for the modelling of negative permeability values using the iteration scheme employed. An advantage of the reduced scalar potential method which must be noted is that the effects of arbitrary current sources which can be modelled using the Biot-Savart equation and reduced scalar potential method are not easily, if at all, soluble using the total scalar potential method.

5 GRAPHICAL DISPLAY OF MAGNETOSTATIC PARAMETERS:

The values of magnetic potential obtained from NASTRAN are calculated at nodes, but the permeability is clearly an element property. The values of B and H are obtained by differentiating the potential, ϕ, and they, and the derived energy density, U, are essentially element data. The module which updates NASTRAN's element stiffness table HKELM outputs to a file a table of element results consisting of the vector components of H and B, in addition to permeability and energy density, which can be directly read by the PATRAN finite element analysis program, and graphically displayed by PATRAN. plates

TABLE 2: CALCULATED AND NASTRAN FLUX LEVELS

MMF (GILBERT)	FLUX (NASTRAN) MAXWELLS	FLUX (CALCULATION) MAXWELLS
20	145	120
50	380	280
100	770	560
150	1160	840
200	1430	1100
300	1659	1200
350	1771	1250
450	1992	1350

7 DERIVATION OF ELECTRICAL PARAMETERS FROM THE FINITE ELEMENT MODEL:

The electrical parameters of interest are the inductance of the circuit, which determines the rate of current rise when the coil is activated, and the current level at which saturation occurs. In a nonlinear magnetic device the inductance varies with current level, and must be measured by a complex incremental method, but the initial inductance is well-defined and readily measured. Familiar formulae for the voltage across an inductor are:

$$V = -L \cdot di/dt \tag{7}$$

$$V = -N \cdot d\phi/dt \tag{8}$$

where V is the potential difference across the coil, i is the current, N is the number of turns and L is the inductance. Combining these equations gives:

$$L = -N \cdot d\phi/di \tag{9}$$

where i may be related to MMF as shown in table (1). The coil has 320 turns, giving an initial permeability value for the device of 9.9 mH, compared with a measured value of 9.2 mH. The variation is well within that to be expected due to assembly variation. By calculating di/dt for different current levels, the current profile for a given applied transient voltage may be calculated using equation (7). Fig.(9) shows a current profile for an applied voltage of 58V using a current-limiting driver circuit. The initial slope corresponds to the measured and calculated values of inductance, while the "knee" in the curve occurs at a current level of approximately 800-900 mA, or an MMF of 256-320 Gilberts. This corresponds well with the "knee" of the curve of Fig.(8), which shows the circuit beginning to saturate at this level of MMF.

(2) through (10), discussed below, are contour plots produced by PATRAN. In interpreting these results it must be noted that PATRAN does not plot element results but instead averages element data at shared nodes. This is correct if there are no discontinuities in the material properties of the model, but produces spurious effects at boundaries between materials of different permeability. The well-behaved slopes at the boundary between steel and air in these figures should in reality be step-function discontinuities. Fig.(7) shows a section through an iron component with magnetic field as calculated by NASTRAN, and the distortion as a result of nodal averaging . So long as the component is more that one element wide the central value will be correct, and will be representative of the value in the elements on either side of it. The distortion in the plot can be reduced by location of very thin elements at the interface, so that the averaging effect will be confined to them, but at considerable cost in model size. The magnetic potential is written to PATRAN in nodal form using NASTRAN's OUTPUT2 module, and is not subject to distortion in plotting.

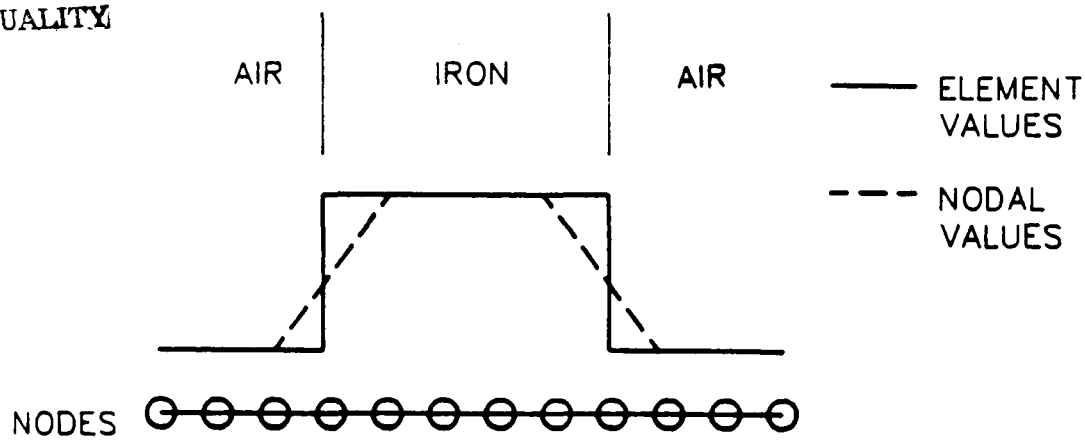


Fig.(7): DISTORTING EFFECT OF NODAL AVERAGING ON PLOTS OF ELEMENT DATA

6 APPLICATION OF THE FINITE ELEMENT METHOD TO THE ROTATING ARMATURE PRINTHEAD:

Modelling the coil as a discrete continuity as described above, there is a significant gain in accuracy to be had by replacing the simple circuit by a finite element mesh consisting of elements representing both the ferromagnetic materials and the surrounding medium. This is particularly advantageous if, as in this case, we are interested in the behaviour of the material in the saturation region (the region of Fig.(3) where the slope of the curve begins to become small) and for complex geometries which cannot be modelled simply. Plate(1) shows a two-dimensional finite element model of the armature and

core. Using single point constraints or loads a discontinuity can be created at the center of the coil and the quantities B, H, V, μ, V and U obtained throughout the model for a range of discontinuity values. The NASTRAN results for this simple geometry are compared with values obtained by a graphical approximation method in Table (2). Fig.(8) is a plot of the curve of the graphical solution with the NASTRAN results superimposed. Both the NASTRAN analysis and the approximate calculation show that the behaviour of the system is linear for MMFs less than approximately 300.0 Gilbert. Contour plots of V, B, H, μ and U show that the subsequent nonlinear behaviour is due to the onset of saturation of the armature. plates (2) to (10) show carpet plots of V, B, H, μ and U for values of MMF below, above and at the knee of the curve of Fig.(8). The results obtained show that agreement is fair between the hand calculation and the NASTRAN model for the unsaturated part of the curve but less good for the saturation region. The variation is not surprising when the crude nature of the hand calculation is considered.

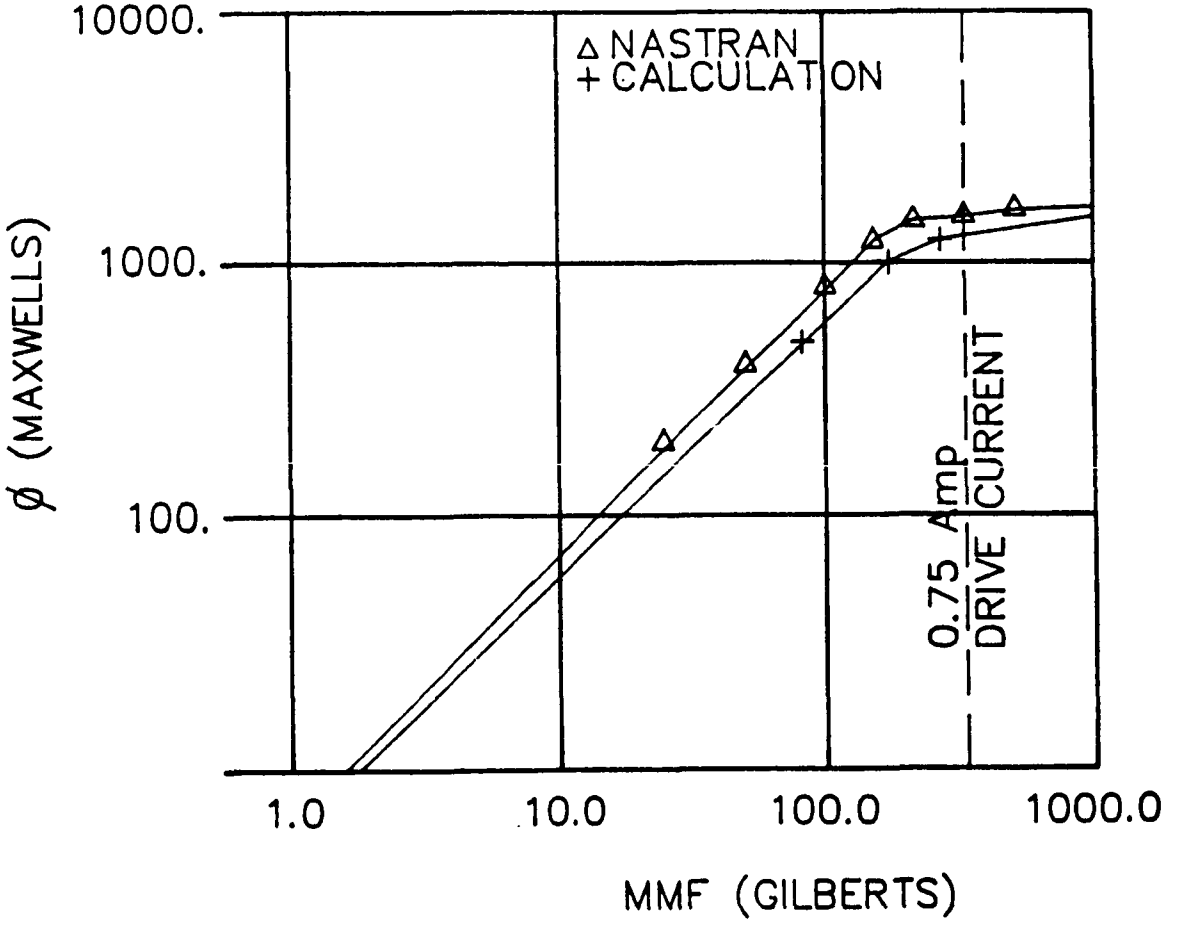
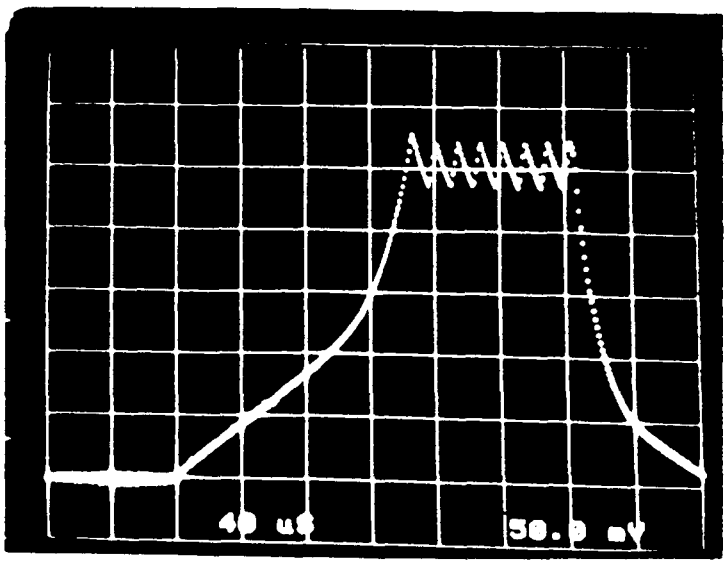


Fig.(8): Calculated and NASTRAN values of magnetic flux vs. MMF



0.5A per division

Fig.(9): Current Profile for printhead coil with 58V Applied EMF

8 CALCULATION OF MAGNETIC TRACTION FORCES:

The finite element model allows calculation of distributed forces on the armature, using the free pole method described in (6). The equation for the normal component of force in SI units is:

$$F = \mu_o \cdot \left(1 - \mu_r^{-2}\right) \cdot H_n^2 \qquad (10)$$

Using this equation, the forces tending to close the air gaps were calculated from element magnetic field values and the resultant force at the armature tip as a function of current plotted against experimental values in Fig.(10). The agreement is surprisingly good in view of the simplicity of the model. A hand calculation based on the total flux in the circuit gave forces about 100% higher than those found using NASTRAN.

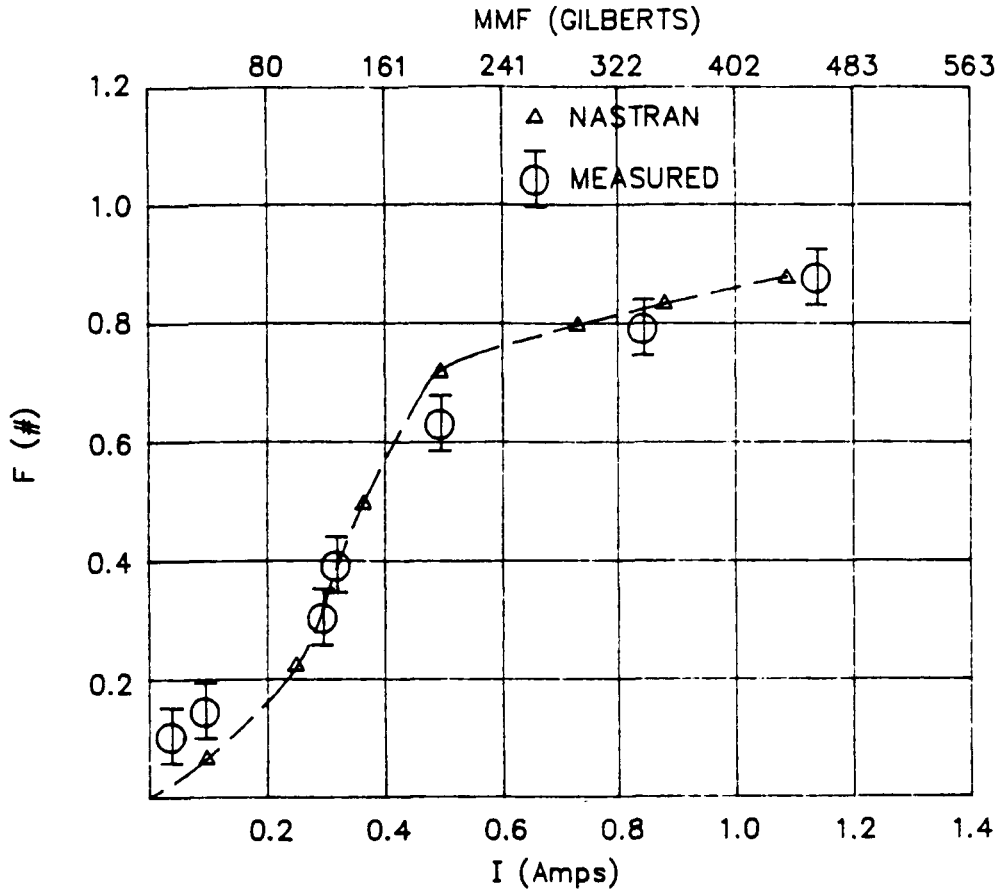


Fig.(10): CALCULATED AND MEASURED FORCES AT ARMATURE TIP

9 DISCUSSION:

The NASTRAN model predictions of the magnetostatic behaviour of the assembly show good agreement with experimental data. Output of magnetic field data to PATRAN allows the visualization of the magnetic field and flux paths within the system. The plots of permeability and of energy density are of particular use in observing the onset of saturation in the iron parts of the circuit, and in comprehending the behaviour of the device. The calculations of magnetic traction force have a degree of accuracy which is unobtainable by conventional methods, and which is surprising in view of the simple nature of the model. Of particular interest is the ability of the model to predict the electrical behaviour of the device, which should allow the use of NASTRAN in conjunction with a circuit analysis package such as SPICE to predict the transient and low frequency behaviour of circuits including nonlinear magnetic devices such as power transformers, relays and other actuator mechanisms.

While it is unlikely that the agreement between NASTRAN and measured data in this case is fortuitous, similar analysis of other magnetomechanical devices is required to fully validate the methods used and determine how generally they may be applied.

10 REFERENCES:

(1). M. M. Hurwitz and E. A. Schroeder, 7th NASTRAN Users' Colloquium, NASA CP-2062, 1978

(2). NASTRAN Users' Manual, NASA SP-222(08)

(3). T.J. Sheerer, 14th NASTRAN Users' Colloquium, NASA CP-2419, 1986

(4). J. Simkin and C. W. Trowbridge, IEE proc. **127**,pt **B**, no 6, 368-374 (1980)

(5). T. W. McDaniel, R. B. Fernandez, R. R. Root and R. B. Anderson, Int. J. Num. Meth. Eng, **19**, 725-737 (1983)

(6). T. J. Sheerer, 15th NASTRAN Users Colloquium, NASA CP-2481, 1987

ORIGINAL PAGE IS
OF POOR QUALITY

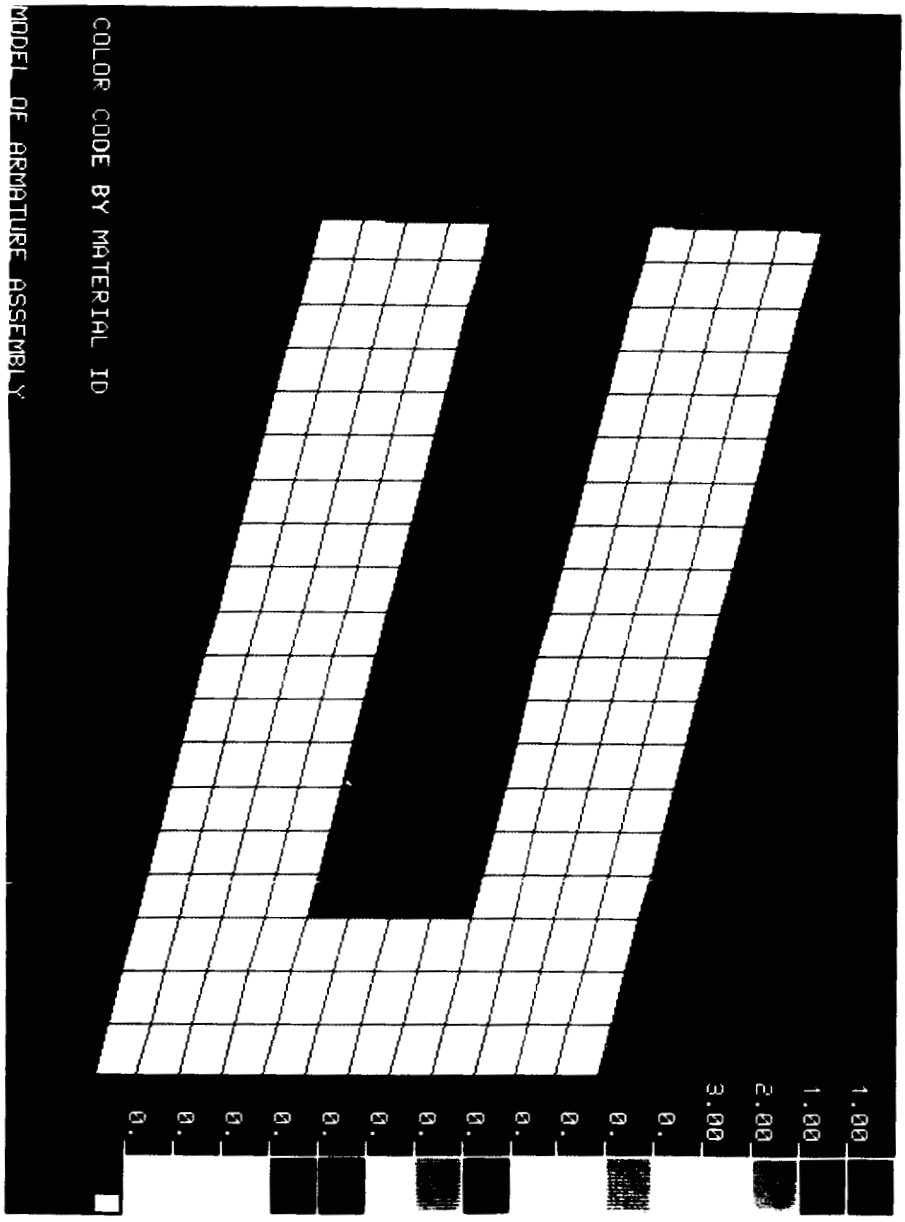


PLATE 1

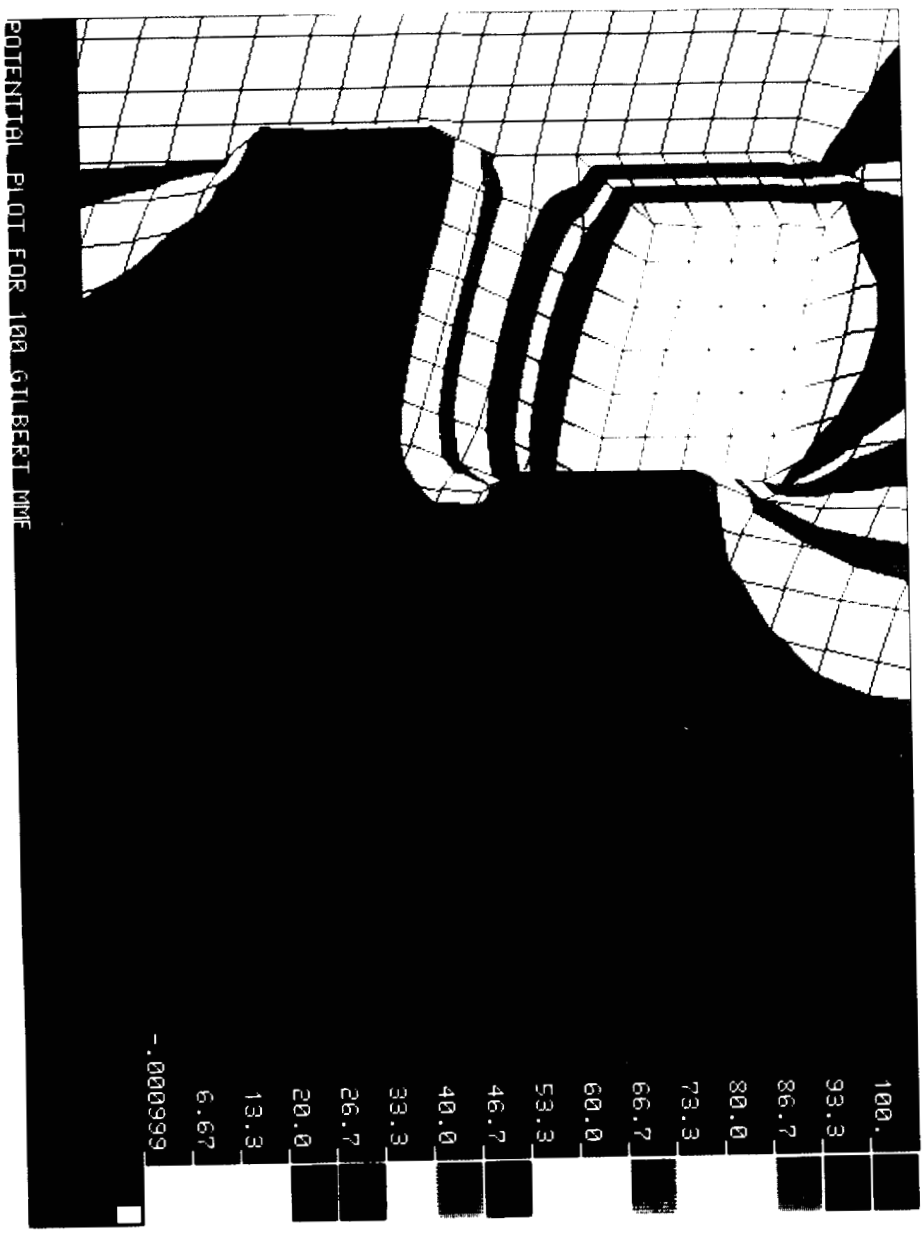


PLATE 2

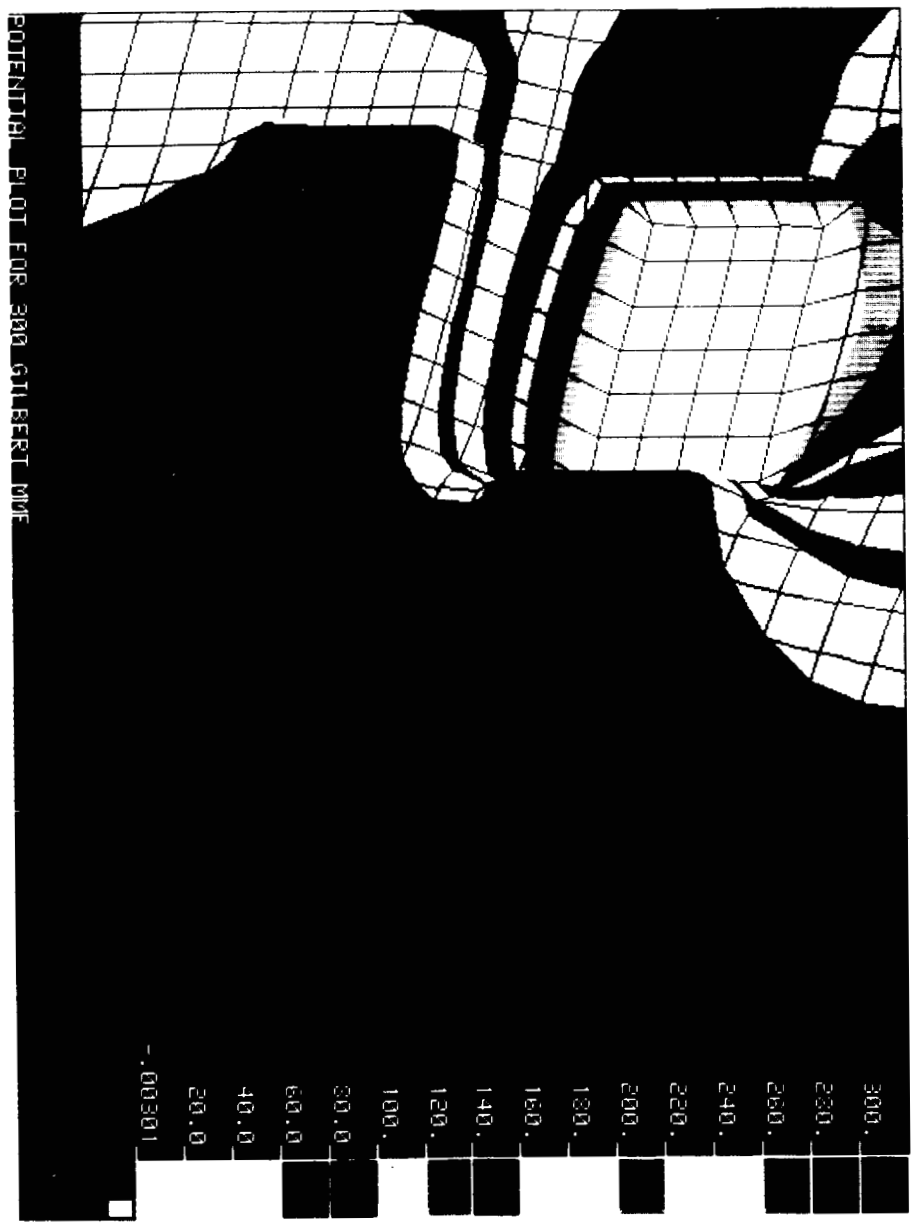


PLATE 3

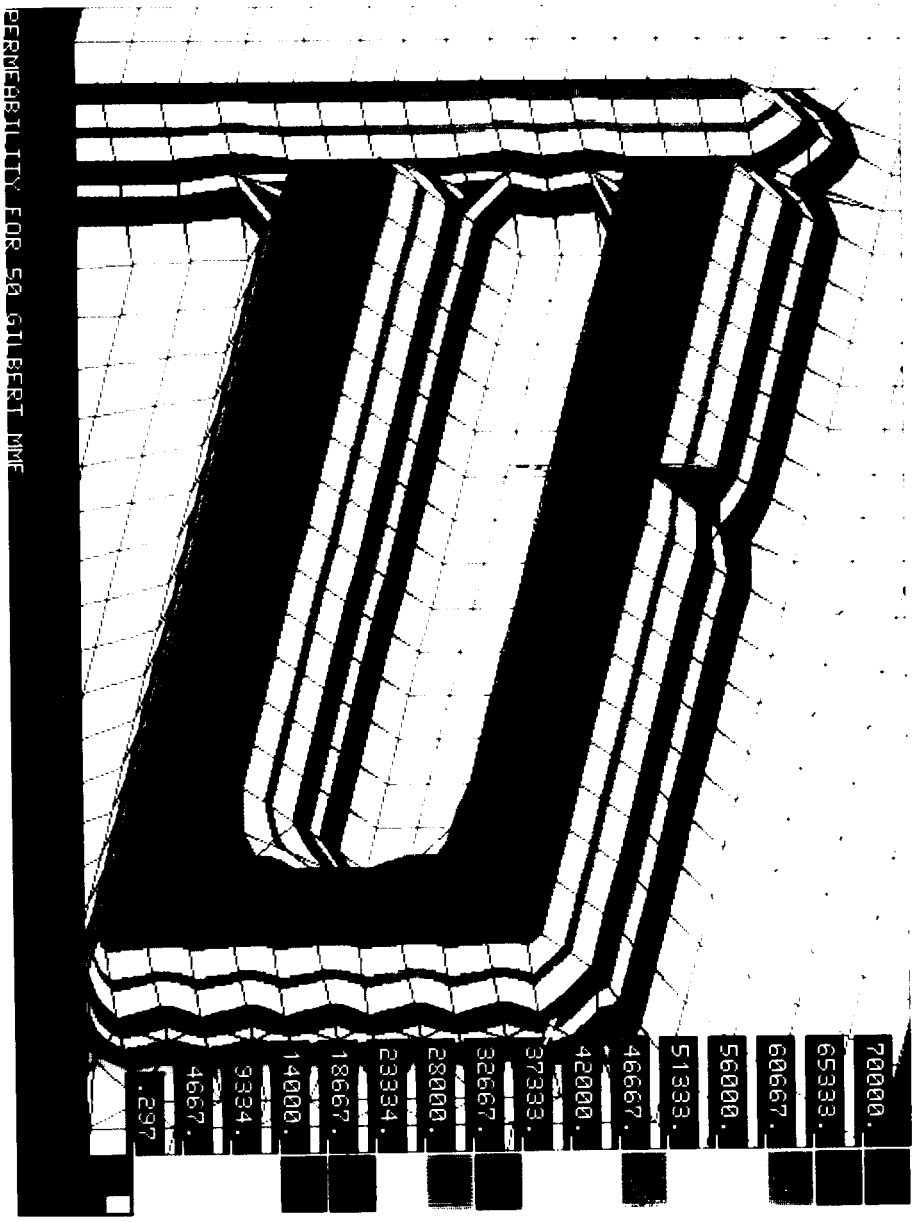


PLATE 4

ORIGINAL PAGE IS
OF POOR QUALITY

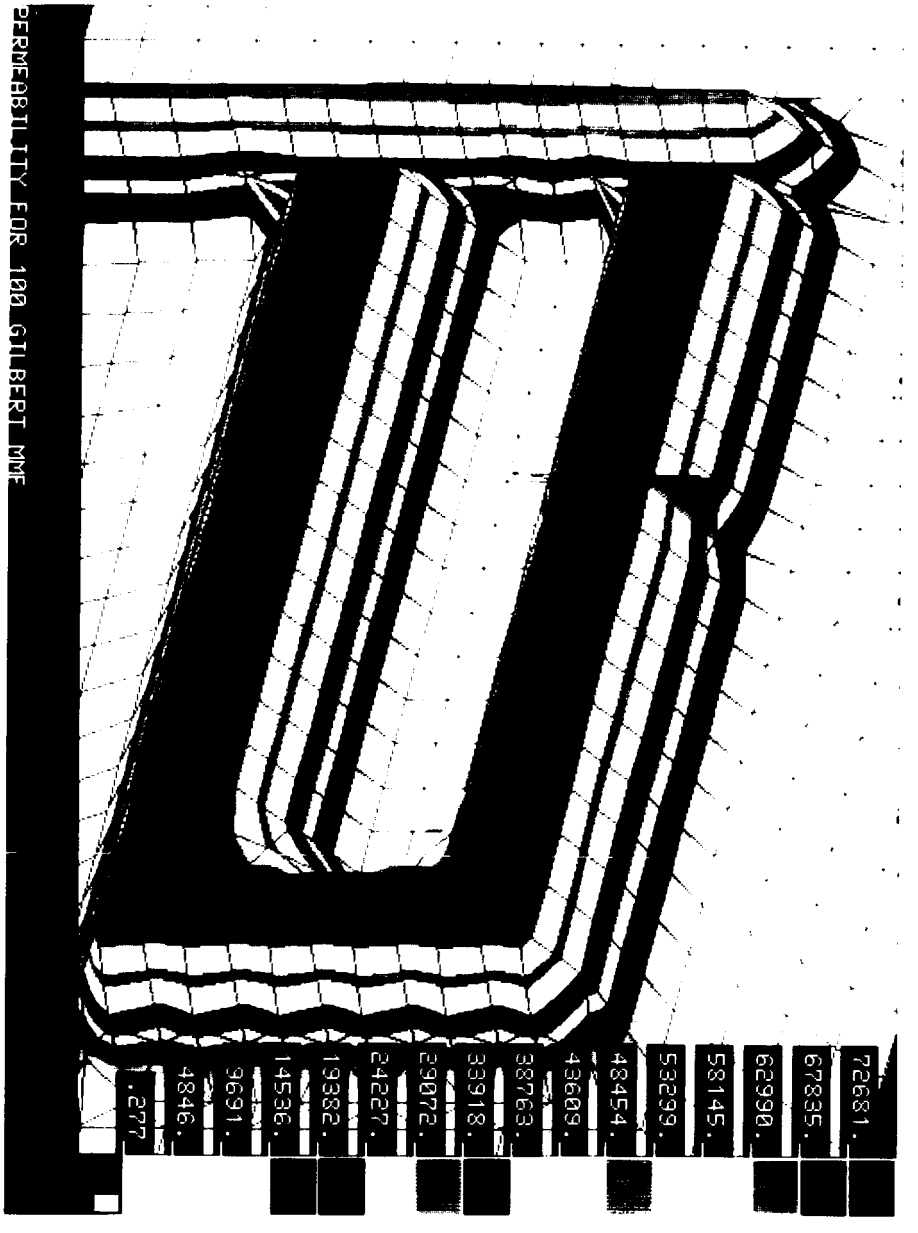


PLATE 5

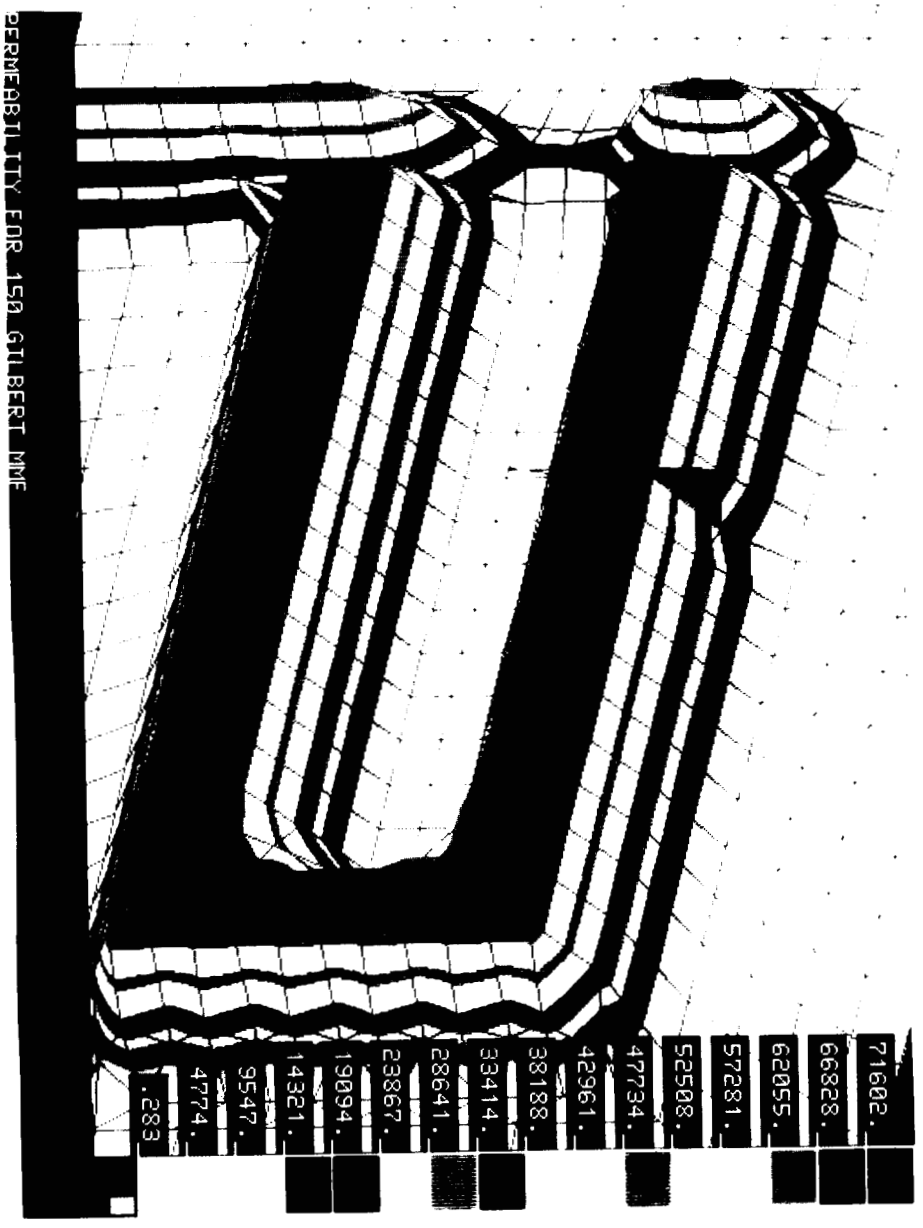


PLATE 6

ORIGINAL PAGE IS
OF POOR QUALITY

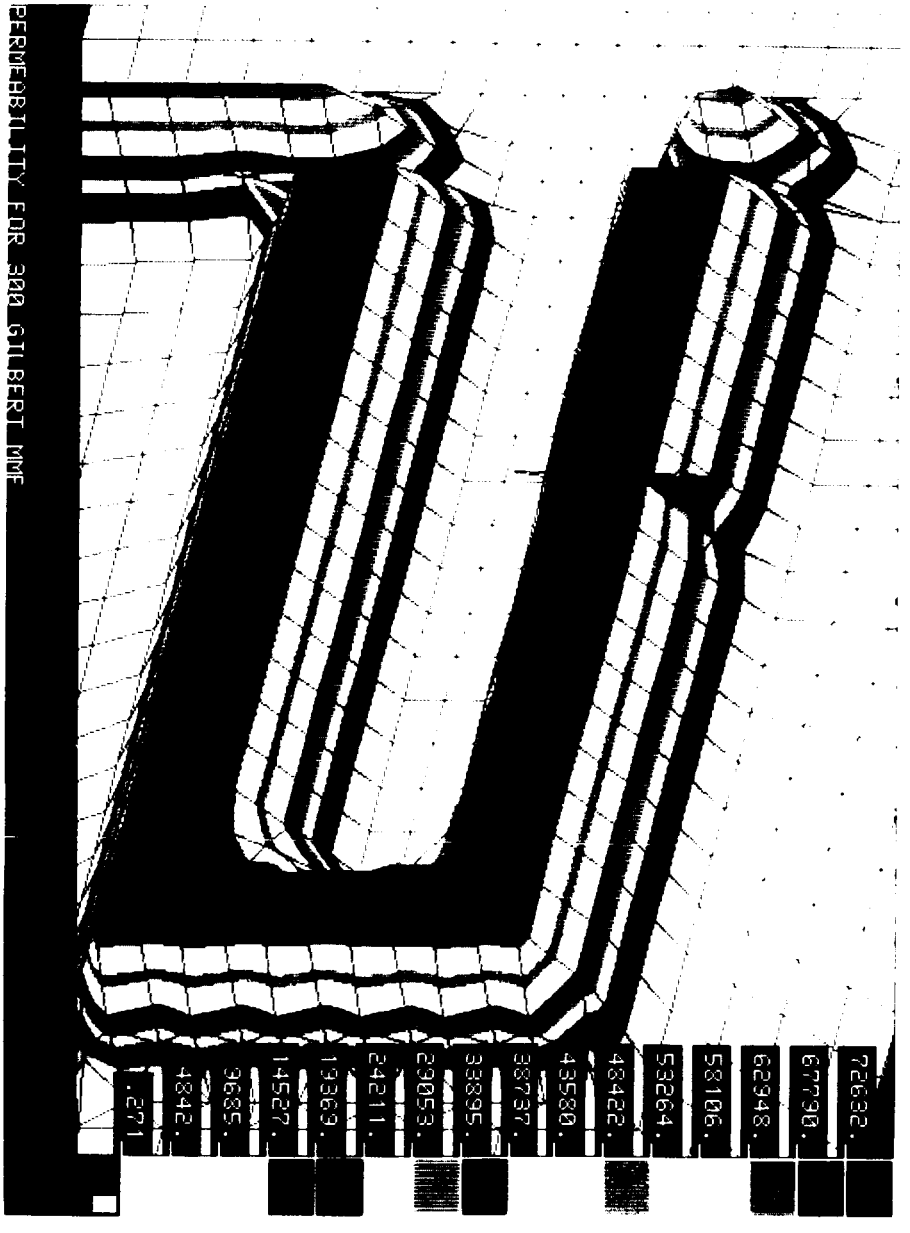
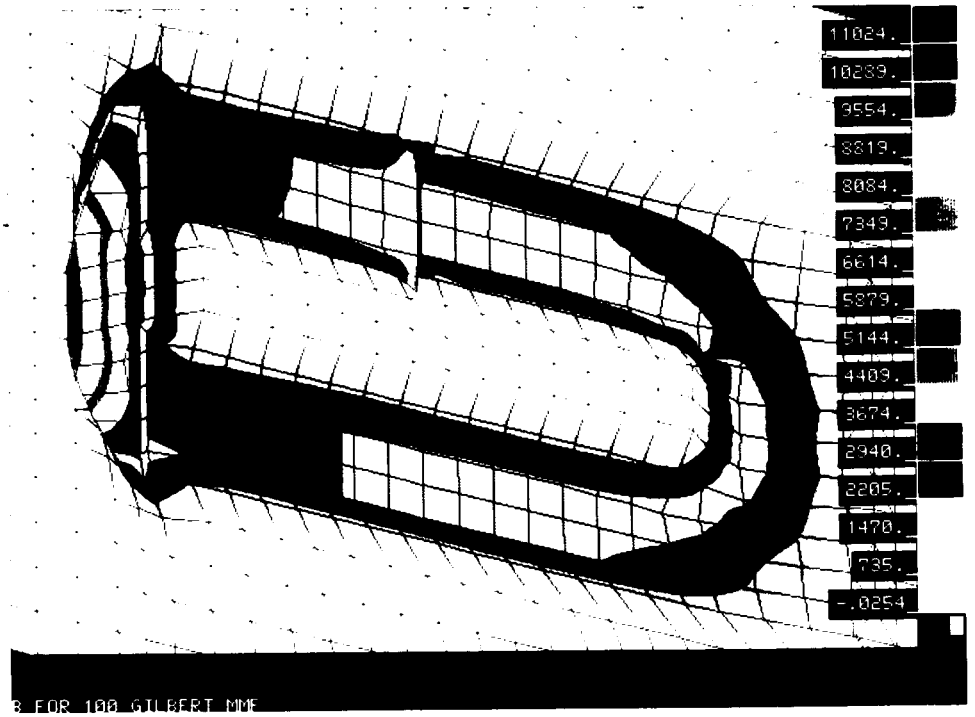


PLATE 7



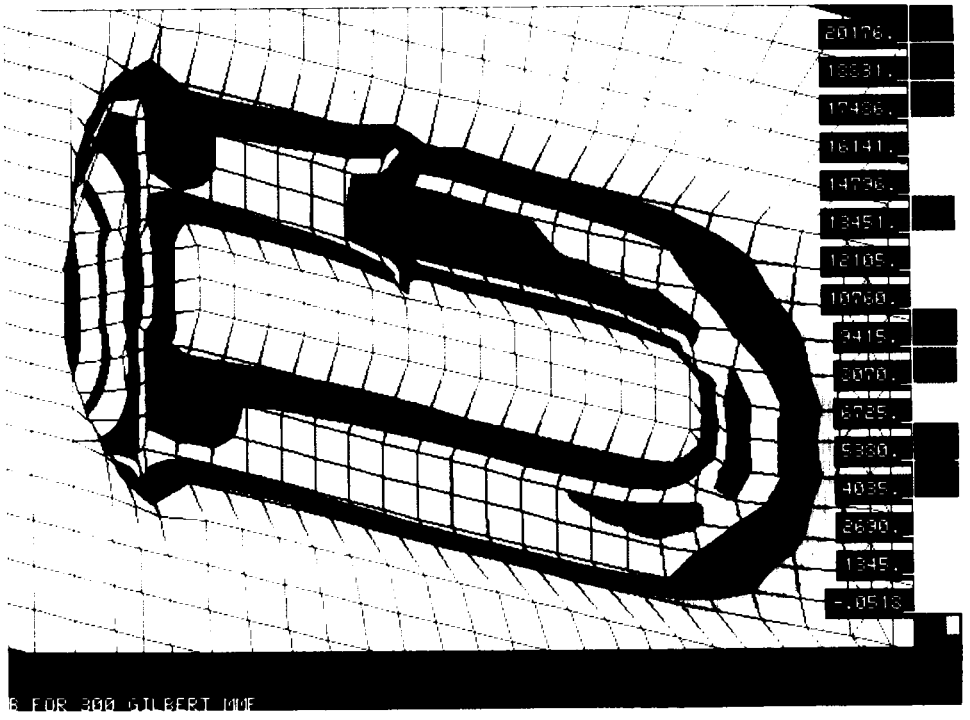
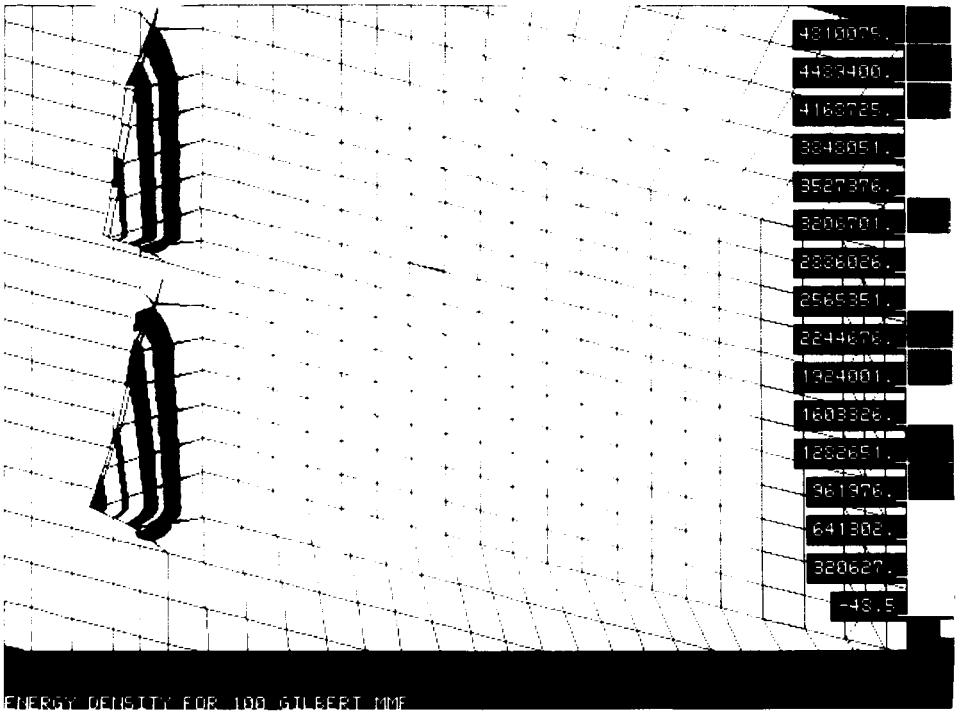


PLATE 8



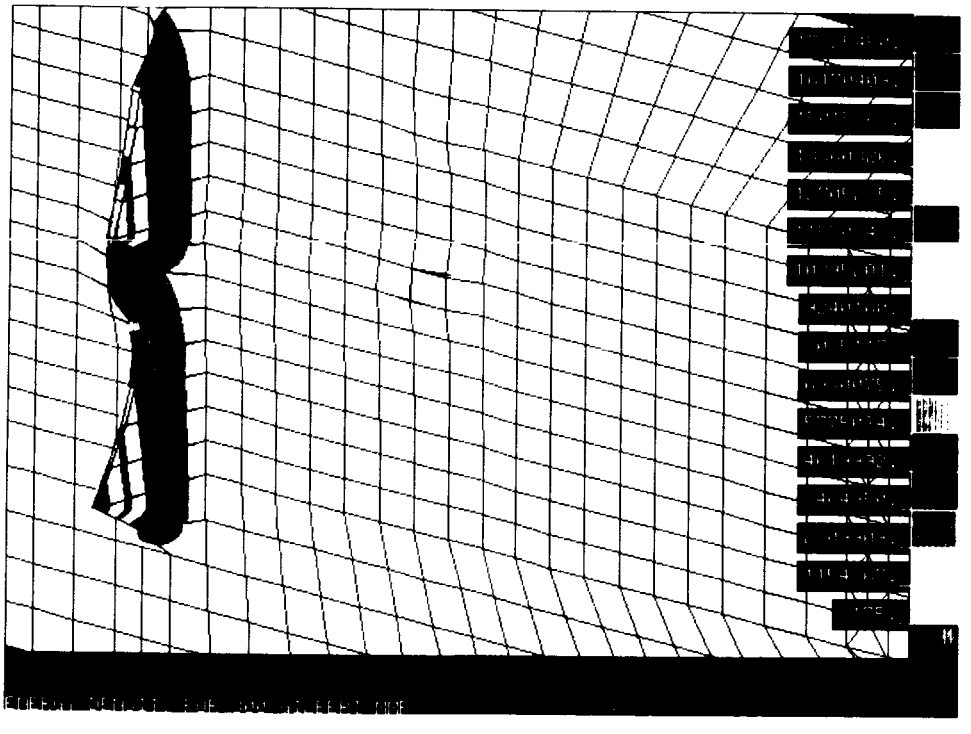
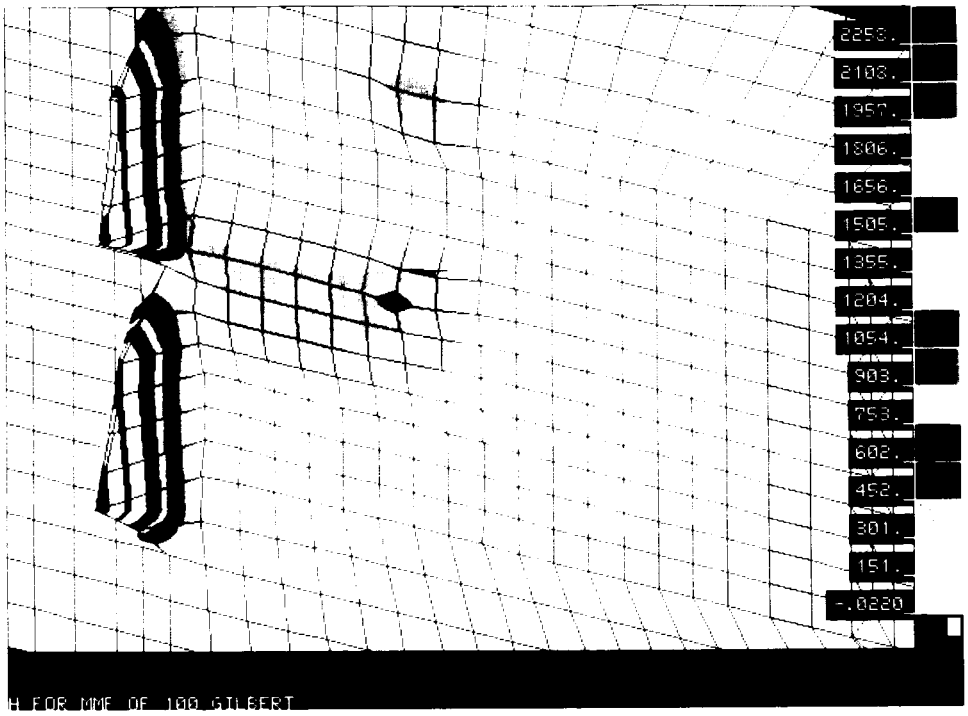


PLATE 9



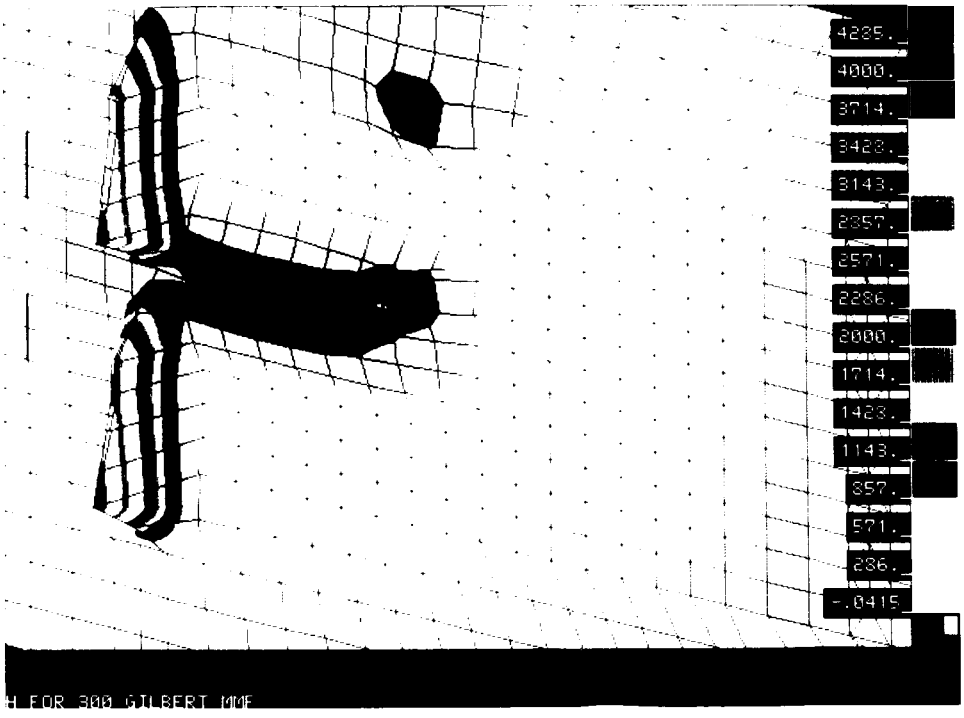


PLATE 10

ARTIFICIAL INTELLIGENCE AND NASTRAN: VE-1 --- AN R.B.E.S. INTRODUCING NASTRAN

V. Elchuri

Aerostructures, Inc.

ABSTRACT

The techniques of Artificial Intelligence are applied in developing a Rule-Based Expert System, VE-1, to provide introduction to NASTRAN.

Although the expertise of VE-1 is scoped, by design, to address the introductory phase of learning about NASTRAN, the methods employed are applicable in developing specialized experts in the many specific areas of NASTRAN.

INTRODUCTION

Although NASTRAN is described as a computer program for the solution of a variety of structural problems by the finite element method, it would not be an overstatement to also describe it as an engineering discipline in itself.

Beginning with the times several years ago, when the original specifications for NASTRAN were developed by the NASA's Ad Hoc Group for Structural Analysis, comprising distinguished visionaries, to the present day with innumerable users around the globe -- the NASTRAN "system" has grown manifold. A good part of this growth can be directly attributed to the numerous and varied applications the Industry has found for NASTRAN in seeking practical solutions to real problems. And rightly so, it is not surprising that lately, NASTRAN has become the subject of regular coursework at many of the Universities across the country.

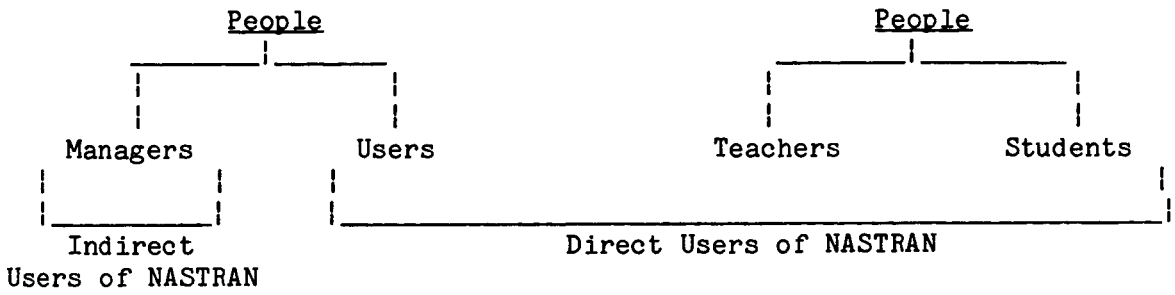
Figure 1 illustrates the most common use of NASTRAN in the Industrial and University environments. While the Industry is heavily oriented towards NASTRAN applications, the majority of the student projects at the Universities contributes to the more fundamental aspects of NASTRAN. The figure also observes the types of people that directly or indirectly interact with NASTRAN.

This paper is presented with a view to efficiently and economically facilitate the "Introduction of NASTRAN" on the part of managers training new users, on the part of self-motivated managers upkeeping themselves, on the part of teachers educating students, and on the part of old users helping develop new users.

This paper is also presented to simultaneously and appropriately bring the methods from the discipline of Artificial Intelligence to innovatively address the task of introducing NASTRAN.

FIGURE 1. USE OF NASTRAN IN INDUSTRY AND UNIVERSITY

	INDUSTRY Activities	UNIVERSITY Activities
Most likely sequence of interests in NASTRAN use ↓	1. Applications 2. Local Tailoring/Mods (I/O, Plotting, etc.) 3. Moderate-sized Specific Developments/Procedures (DMAP's, e.g.) 4. Research leading to fully developed new capabilities	1. Research in Specific areas Student Projects in, e.g., - Finite Element development - Mathematical/Numerical aspects of selected solution algorithms (e.g., Eigenvalue extraction, numerical integration, etc.) 2. Applications 3. Local Mods (I/O, Plotting, etc.) 4. DMAP'S 5. Fully developed new capabilities



Of the many areas associated with the field of Artificial Intelligence -- such as Natural Language Interfaces and Understanding, Symbolic Mathematics, Robotics, and Modeling of Human Problem-solving (Reference 1) -- this paper deals with the area of Knowledge Engineering, in general, and Expert Systems, in particular.

The Expert Systems are examples of the practical applications of the research conducted in Artificial Intelligence (Reference 2). Such systems embody knowledge of a specific application area, and employ inference mechanisms to utilize their knowledgebase in suggesting solutions to problems in the specific area of application. Medical diagnosis, mineral prospecting, chemical structure elucidation, and computer-system configuration are some areas wherein the Expert Systems are currently being utilized.

Rule-Based Expert Systems (R.B.E.S.'s), as the name implies, operate on a collection of facts and rules involving these facts. Some R.B.E.S.'s are also designed with the capability to "learn" more facts as they function.

VE-1 is an R.B.E.S. designed to perform the task of introducing NASTRAN. Some of its design features are also presented in this paper.

With a view of wider applicability and utility, VE-1 is presently designed for personal computers.

It is expressly intended of VE-1 to systematically and substantively impart necessary and sufficient knowledge to the user enabling him/her to

1. Be informed of NASTRAN,
2. Get to know about its capabilities and applications,
3. Learn about its Documentation/Manuals,
4. Recognize and understand a typical NASTRAN printout in terms of its organization, and most of the commonly used terms,
5. Effectively and efficiently search for information in NASTRAN Manuals,
6. Be informed about other avenues regarding learning about NASTRAN, and
7. Finally, apply NASTRAN to solve his/her structural analyses problems, and
8. (Maybe!) start thinking about new and varied and enhanced applications of NASTRAN.

VE-1 AND NASTRAN

Defining VE-1's "Expertise"

In order to define and establish the scope of applicability and limitations of VE-1, the following general requirements are specified for its "expertise," i.e., its knowledgebase (facts) and inference mechanisms (rules):

1. Sufficiently knowledgeable about NASTRAN. The knowledge is categorized and stored, for instance, in terms of answers to questions like,

   a) What is NASTRAN?
   (A finite-element based computer program to solve structural analysis problems, etc.)

   b) How to use NASTRAN?
   (User typically prepares one data file consisting of three parts - Executive, Case Control and Bulk Data Decks, etc.)

   c) Where to read about NASTRAN?
   (NASTRAN Manuals - Theoretical, etc. (References 3-6); NASA/COSMIC Annual Colloquia Proceedings; NASTRAN User's Guide, etc.)

   d) How to read about NASTRAN?
   (Especially in view of the NASTRAN Manuals spanning over half-a-dozen thick volumes.
   - o VE-1 - one approach
   - o COSMIC seminars, etc.)

2. Capable of understanding the User's questions at hand and needs in general.

   Capability to provide relevant parts of stored knowledge as answers to User's specific questions.

   At the same time, capability to provide guidance in case the User doesn't know what to ask.

3. Capable of disseminating/imparting information/knowledge in a systematic, organized and patient way by providing well-balanced information, and references and cross-references.

4. Capable of informing the User if the answer(s) to his/her specific inquiry is not available based on the current knowledge, and capable of learning. For instance,

   VE-1: "My present knowledge does not have an answer to your specific query.

   If you would like to enhance my knowledge now, please indicate YES or NO.

   User: YES

   VE-1: "Please type in your information/answer to your specific query.

   Use F6 key to indicate you are done.

   or

User: NO

VE-1 returns to previous screen/level/window from where a branch was taken to end up with the no knowledge situation.

5. Capable of helping the User by allowing him/her to add supplementary/footnote information for subsequent reference and convenience.

Modes of VE-1 to Learn About NASTRAN

Due to the facts

1. that people wishing to learn NASTRAN probably come from a wide spectrum of education and experience,

2. that individuals have preferential, personalized learning habits and speeds, and

3. that learning is a progressive and iterative process,

VE-1 has been designed to facilitate learning about NASTRAN in three convenient Modes:

Mode 1: General Learning about NASTRAN,

Mode 2: Learning by References and Cross-References within NASTRAN Manuals, and

Mode 3: Learning by Specific Examples.

Each of these Modes is further discussed in the following Sections. Examples of the possible interactive conversations between VE-1 and the User are given to illustrate the learning process.

Mode 1: General Learning About NASTRAN

Upon beginning to use VE-1, an example of the first level of screen to appear is illustrated in Figure 2.

User's selection of Mode 1 of learning about NASTRAN, and his/her subsequent selections (as indicated below) lead to the following interactive conversation.

(In order to follow the conversation conveniently, and for the convenience of program development, a number code appears in front of the statements made by VE-1 on every screen level. As an example, consider

FIGURE 2. EXAMPLE OF A FIRST LEVEL SCREEN IN VE-1

VE-1

VE-1: Hello, this is VE-1, your Rule-Based Expert System for an Introduction to NASTRAN.

You can learn about NASTRAN in any of the following three Modes.

Mode 1: General Learning about NASTRAN.

Mode 2: Learning by References and Cross-References within NASTRAN Manuals, and

Mode 3: Learning by Specific Examples

Indicate your choice of Mode number, or type QUIT to leave VE-1.

User: 1

1.2.4 VE-1: ... .

1 is the indication of learning Mode No. 1 by VE-1.
2 is the option no. selected by User, after VE-1 enters Mode 1.
4 is the option no. selected by User, following VE-1's response to his/her option no. 2.)

(New Screen Level)

1. VE-1: Do you know anything about NASTRAN?
(Select 1, 2 or 3)

1. Yes
2. No
3. Not sure/Maybe/Don't know/...Mmm...

User: 1

1.1 VE-1: How much do you know about NASTRAN?

1. A little/Not too much/Just heard about/...Mmm...
2. Working knowledge/Used before/Familiarity
3. A good deal

User: 1

1.1.1 VE-1: ...

(Conversation would be similar to that discussed for Response Codes 1.2 and 1.3, i.e., when the User answers 2 or 3 for the Question 1.)

1.1.2 VE-1: Are you looking for any specific information re. NASTRAN?

1. Yes
2. No
3. Not sure

User: 1

1.1.2.1 VE-1: Because of your working knowledge and familiarity with NASTRAN,

1. This mode of learning and seeking information may be too elementary for you.

2. You may do well by following one or both of the other two informatory/learning modes of VE-1, i.e.,

   - Learning by References and Cross-References within NASTRAN Manuals (Mode 2), and
   - Learning by Specific Examples (Mode 3).

3. To do this, use ESC key to get back to the first screen, and then select Mode.

4. Otherwise, use F1 key to get back to the previous screen, and continue.

User: F1 key

1.1.2.2 VE-1: Given your working knowledge and familiarity with NASTRAN, and the fact that you are not looking for any specific information,

1. VE-1 may not be of help/use to you at this time.
2. Use ESC key, and then opt to quit VE-1.
3. If not, use F1 key to go back and continue.

User: F1 key

1.1.2.3 VE-1: If you are not sure whether you need any information at this time,

1. Use ESC key, and then choose to quit VE-1.
2. Or perhaps, after the ESC key, opt for Mode 2 of learning more about NASTRAN using VE-1's inherent unification and backtracking capabilities.
3. Alternatively, F1 key will take you back if you wish to continue.

User: F1 key

1.1.3 VE-1: Although VE-1 is an Expert System, it's principal purpose is to address the introductory phase about learning NASTRAN.

With your extensive knowledge of NASTRAN,

1. VE-1 may not be of use to you.
2. Use ESC key, and select to quit.
3. Or, use your expertise to explore/critique VE-1; After ESC key, select any of the three learning modes and continue.

4. You can also go back using F1 key.

User: ESC key

1.2 VE-1:
1.3

1. NASTRAN is NASA's STRuctural ANalysis computer program.

2. It is a large and comprehensive program capable of addressing a variety of structural problems.

   Examples of NASTRAN's analytical capabilities include (A complete list is given later)
   - Static analysis
   - Static analysis with Inertia Relief
     (Relief in applied loads due to the free-body inertia of the structure)
   - Static analysis with differential stiffness
     (Incremental stiffness due to deformation of the structure under applied loads)
   - Dynamic analysis
     (Modal analysis, Response analysis, in frequency and time domains)
   - Heat transfer analysis
     (conduction, convection, radiation)
   - Aeroelastic analysis
     (Flutter, Forced response)
   - Acoustics analysis

   (Use F2 key to go forward, F1 key to go back)

---

3. NASTRAN is based on the finite element method of structural analysis.

   The space occupied by the material of the structure is divided into a finite number of elements.

   Based on the geometry of the structure, and the problem of interest, the elements can be linear (bars, rods, beams), surface-type (plates, shells) or volume-type (solids).

   The corners of element boundaries are called the Grid points. The "flexibility" of the structure manifests through the "degrees of freedom" assigned to the Grid points.

   In structural problems, the degrees of freedom can be the translational and rotational motion of the structure at the Grid points.

In heat transfer problems, the degrees of freedom can be the temperature at the structure's Grid points.

In acoustics problems, acoustical perturbation pressures can be the degrees of freedom at the Grid points of the structure.

(Use F2 key to go forward, F1 key to go back)

And so on, the informative/educative process continues in Mode 1. Some of the important views kept in mind in designing VE-1 are,

1. The informative aspects are appropriately organized in order to give the new user a sequential flow of information. (The flexibility to go back and forth with the touch of F1 and F2 keys is extremely useful in the learning process.)

2. The information is kept brief. This is to keep the user on top of things at all times. It is felt that this advantage decisively outweighs the disadvantage of not informing the user of all details. It is significant to monotonically raise the confidence level of the user in learning about NASTRAN. With a new user, the educative process can be termed successful, if he/she becomes informed/knowledgeable to the point of getting the details freely, for instance, from the NASTRAN Manuals.

3. From the viewpoint of completeness of information, especially with regard to that derived from the NASTRAN Manuals, the Mode 2 of VE-1 has been specifically designed to accomplish this. This is discussed further next.

Mode 2: Learning by References and Cross-References within NASTRAN Manuals

As discussed earlier in this paper, the education and experience of people getting to be introduced to NASTRAN cover a lot of ground. Coupled with the individual's learning methods and habits, it is not too difficult to surmise that there is no one efficient and adequate method of introducing NASTRAN -- and hence the three Modes of VE-1.

The singular principal reason for creating Mode 2 of VE-1 is that the four NASTRAN Manuals contained in about 6 or more voluminous books/folders are too difficult to follow, comprehend and grasp. This is not a criticism of the Manuals in any way -- for a program, or better yet a system, of the magnitude and versatility of NASTRAN, it would take all the pages of all the Manuals to justifiably document NASTRAN. But when it comes down to a user -- a new user at that -- the size of the documentation does not help -- nor does it suggest where and how to begin.

Experienced users, who have learned their way both by association with other (previously) experienced users and try-and-learn opportunities, would

almost always find the NASTRAN Manuals to be extremely and routinely useful in their practice.

The utility of Mode 2 of VE-1 is, by design, to both the new and the experienced user.

Mode 2 builds up on the excellent method employed in the Demonstration Problems Manual section on "Demonstrated Features of NASTRAN," beginning on page 5 (Reference 6).

The A through I categories of the NASTRAN features demonstrated in the Manual, starting with Physical Problems and Solution Methods to Execution Options and Output Options cover the entire spectrum of analytical capabilities offered by NASTRAN. The further expansion of each of these categories into the next level of logical sub-categories described on the subsequent pages of the Manual systematically and almost completely demonstrate the breadth of NASTRAN's versatility.

Mode 2 of VE-1 picks up from every one of these 134 sub-categories and establishes their individual relationships with the corresponding and relevant information from each of the NASTRAN Manuals -- the Theoretical, User's, Programmer's and Demonstration Manual. (The Demonstration Manual is included not only for completeness of reference and cross-reference information, but also to accommodate the Demonstration examples created since the manual's publication -- which have not yet found their way into the manual, but are available to User's on tape for actual running.)

Presently, the design of VE-1 calls for the reference and cross-reference information to be limited to identifying to the User the section (or sub, or sub-sub, etc.) numbers and titles along with the Manual, for each of the NASTRAN Manuals. This approach is felt to be sufficiently adequate to allow and guide the User across the Manual boundaries to conveniently and completely seek the information of interest.

An example of VE-1's Mode 2 design in helping the user learn about scalar elements in NASTRAN is illustrated in Figure 3. It is also noted that for the scalar elements information in the User's manual, references to the page numbers of the relevant Bulk Data cards are implicitly necessary for the completeness of help.

The last of the three Modes of VE-1 is discussed next.

Mode 3: Learning by Specific Examples

The purpose of this Mode, as the name indicates, is to help the User learn how to prepare for and conduct a structural analysis problem using NASTRAN.

For brevity, only the highlights of the Mode 3 design are described.

FIGURE 3. EXAMPLE OF VE-1 MODE 2 FOR REFERENCING AND CROSS-REFERENCING ACROSS NASTRAN MANUALS (SCALAR ELEMENTS)

DEMONSTRATION MANUAL

Feature Category: c. Element Types
Sub-Category: 4. Scalar Spring, Mass, Damper

Demo Problem Nos.: 3-8, 7-1, 9-2, 9-4, 10-1, 10-2, 11-2, 11-3

THEORETICAL MANUAL

Section 5.6: Scalar Elements

USER'S MANUAL (VOL. I)

Section 1.3.8: Scalar Elements

Section 2.4.2:

CELASi	pp. 2.4-40 to 2.4-43
PELAS	p. 2.4-227
CDAMPi	pp. 2.4-35 to 2.4-38
PDAMP	p. 2.4-225
CMASSi	pp. 2.4-62 to 2.4-65
PMASS	p. 2.4-241

PROGRAMMER'S MANUAL

Section 8.7: The ELASi, MASSi, and DAMPi Elements

It is assumed that the User has acquired familiarity, if not knowledge, by means of principally Mode 1, and maybe Mode 2 to a limited extent. Specifically, information regarding facts such as various NASTRAN analysis capabilities (Statics, Dynamics, etc.), that a Rigid Format is the means by which NASTRAN conducts the selected analysis, that the user typically prepares one data (submit) file, that this file contains the Executive, Case Control and Bulk Data decks besides JCL, and so on, is familiar to the User.

Mode 3 presents a list of all Rigid Formats. Predefined ALTER packages are not introduced for clarity, at least for the present. Upon User's selection of the rigid format, all of the necessary requirements for the Bulk Data, Case Control and Executive Control decks are brought forth in that order. From an instructional viewpoint, it is assumed that the User is better informed about the structural/mechanical details of his/her problem than knowing how to use NASTRAN to solve his/her problem. Since most of this data is supplied in the Bulk Data deck, Mode 3 starts with the Bulk Data. Case Control requirements are the next level of details addressed by VE-1. For the Executive Control deck, the user's data are primarily limited to the TIME card.

It is anticipated that the User may, with some experience, become adept enough about NASTRAN to use the Mode 3 of VE-1 to essentially checklist his data.

VE-1 PROGRAMMING LANGUAGE

Due to the inherent characteristics of Rule-Based Expert Systems for internal deductive reasoning, and the designated preference to make VE-1 operational on personal computers, Turbo Prolog (Reference 7) -- a fifth-generation computer language -- was selected to implement VE-1.

Turbo Prolog is a declarative language (Reference 7). This is to say that given the facts, the rules and the goal(s) of an application, the how to accomplish the goal(s) is internally determined. However, it would be incorrect to observe that this internal deductive reasoning is beyond the control of the programmer. The correct way to take note of this fact is that the Turbo Prolog language aids the programmer by relieving him/her from having to write a significant number of intermediate programming steps to accomplish an objective.

Another salient feature afforded by this declarative language for VE-1 is the ability to find all possible solutions to a specific problem in case not all of the variables of the problem are specified. This is quite unlike the traditional programming language like FORTRAN (which is procedural) wherein all information on the right-hand side is necessary before the left-hand side (solution) is determined.

This fact is significantly fundamental to the Mode 2 of VE-1, wherein the users would almost always benefit from VE-1's responses when they query VE-1 with incomplete information.

CONCLUDING REMARKS

It is hoped that VE-1 -- employing the techniques and philosophy of the advancing and maturing discipline of Artificial Intelligence -- would serve a large number, and a wide variety, of both present and future NASTRAN enthusiasts.

In keeping with the growth trends and potentials of NASTRAN, its users, and their knowledge -- the open-ended (updateable) structure of VE-1 has been designed to accommodate user-generated supplementary comments and footnotes for subsequent reference and convenience.

The generics of the approach and methods describing the design of VE-1 are applicable, and considered useful, toward the design and development of other Expert Systems specializing in areas such as Analysis of Structures using Cyclic Symmetry, Substructure Analysis, Aeroelastic analyses, Acoustics analysis, and Heat Transfer Analyses.

Creation and release of Expert Systems along with NASTRAN and its manuals would be useful not only to train new people but also as quick reference for practicing NASTRAN users.

Expert System is not a total replacement for human experts, for the RBES will always lag the human mind that created it in the first place.

Finally, it is felt that a systematic and sustained development of a series of well-thought-out and well-tailored Rule-Based Expert Systems for NASTRAN would help collect, organize and disseminate the expertise of practicing NASTRAN experts for the benefit of NASTRAN and its users for a long time to come.

REFERENCES

1. Winston, Patrick Henry, Artificial Intelligence, Second Edition, Addison-Wesley Publishing Company, 1984.

2. Hayes-Roth, F., Waterman, D. A., and Lenat, D. B., (Editors) Building Expert Systems, Addison-Wesley Publishing Company, 1983.

3. NASTRAN Theoretical Manual, NASA SP-221(06), January 1981.

4. NASTRAN User's Manual, Volumes I and II, NASA SP-222(08), June 1986.

5. NASTRAN Programmer's Manual, NASA SP-223(05), December 1978.

6. NASTRAN Demonstration Problem Manual, NASA SP-224(05), Reprinted September 1983.

7. Turbo Prolog - The Natural Language of Artificial Intelligence, Borland International, Inc., California, April 1986.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CP-2505		

4. Title and Subtitle	5. Report Date
Sixteenth NASTRAN® Users' Colloquium	March 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address Computer Software Management and Information Center University of Georgia Athens, GA 30602	11. Contract or Grant No.
	13. Type of Report and Period Covered Conference Publication
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Also available from COSMIC, Athens, GA 30602

16. Abstract

This document is the proceedings of a colloquium and contains technical papers contributed during the Sixteenth NASTRAN® Users' Colloquium held in Arlington, Virginia on April 25 to 29, 1988. The authors review general application of finite element methodology and the specific application of the NASA Structural Analysis System, NASTRAN, to a variety of static and dynamic structural problems.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
NASTRAN Structural Analysis Structures Finite Element Analysis Colloquium	Unclassified - Unlimited Subject Category 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	189	A09

NASA FORM 1626 OCT 86

NASA-Langley, 1988

For sale by the National Technical Information Service, Springfield, Virginia 22161-2171

C-3