NASA Contractor Report 3038

NASTRAN General Purpose Interface Requirements Document

W. Keith Brown, Frank J. Douglas,
John R. McDonough, and David J. Ziobro

CONTRACT NAS1-12969
AUGUST 1978

NASA

NASA Contractor Report 3038

NASTRAN General Purpose Interface Requirements Document

W. Keith Brown, Frank J. Douglas,
John R. McDonough, and David J. Ziobro
Computer Sciences Corporation
Hampton, Virginia

Prepared for
Langley Research Center
under Contract NAS1-12969

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1978

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

1. INTRODUCTION

1.1 PURPOSE

This NASTRAN (NASA STRuctural ANalysis) General Purpose Interface Requirements Document (IRD) defines standards for deliverables required of New Capability Contractors (NCCs) and relates these deliverables to the software development cycle. It also defines standards to be followed by NCCs for adding to and modifying the code in the NASTRAN software system and for adding to and modifying the four official NASTRAN manuals: The NASTRAN Theoretical Manual, The NASTRAN User's Manual, The NASTRAN Programmer's Manual, and The NASTRAN Demonstration Problem Manual. It is intended that this General Purpose IRD shall be incorporated by reference in all contracts for a new NASTRAN capability.

1.2 SCOPE

There will be two types of IRD: General Purpose and Special Purpose. This document is the General Purpose IRD. There should also be a Special Purpose IRD for each new capability contract. The Special Purpose IRD would define such specifics as:

1. Specifications of the file structure required for deliverables supplied on magnetic tape
2. Range of error message numbers that may be used
3. Particular DIAG numbers that may be used
4. New NASTRAN manual section numbers that may be used
5. Allocation of cells in the /SYSTEM/ CØMMØN block
6. Versions of compilers, assemblers, and operating systems to be used
7. System tests that are NCC-specific

The standards in this General Purpose IRD shall apply to all NCCs; it is not intended, however, to be a detailed tutorial for programming NASTRAN.

1.3 DOCUMENT OVERVIEW

Section 1 of this document is an introduction. Section 2 discusses NCC deliverables and relates these deliverables to the software development cycle. Section 3 gives a brief overview of NASTRAN documentation; defines NASTRAN documentation format rules, including detailed specifications for typists and standards for flow chart symbols; and concludes with a few comments on style and editing. Section 4 gives general standards for adding to and modifying the code in NASTRAN and defines general NASTRAN programming standards.

Section 3, Adding to and Modifying NASTRAN Documentation, was written by Ms. Charlene H. Welch. Ms. Mary L. Edney was responsible for the layout of the document.

2. DELIVERABLES

2.1 NEW CAPABILITY DEVELOPMENT CYCLE

Two objectives of the NASTRAN Contract Monitor (NCM) are to develop a new NASTRAN capability on time and within budget limitations and to provide new capability to the user community in a NASTRAN level that (a) contains the new capability and (b) maintain all previously existing capabilities at a degree of reliability equal to or greater than the previous level. To help meet these objectives, it is necessary that the NCM establish rules and standards for the software development cycle. This cycle, which hereafter will be called the new capability development cycle, includes: defining the specific requirements the new capability software must satisfy; specifying the software's design, programming, and testing; installing the new capability software into the current in-house version of NASTRAN; certifying that the version into which the new capability software has been installed not only fulfills the new requirements but also maintains, at a minimum, existing capability at the previous degree of reliability; and the NCM's accepting this new product and other deliverable items.

The new capability development cycle is simply a sequence of interrelated activities leading to the fulfillment of the NCM's objectives. The periods of time during which these activities take place are called phases. Although there may be overlaps in time between these sequential phases, it is assumed in this section that no overlap exists.

The new capability development cycle is divided into the following six phases:

1. Definition Phase
2. Design Phase
3. Programming Phase
4. System Test Phase
5. Installation Phase
6. Acceptance Phase

Associated with each phase are one or more deliverables. A phase is complete only if the deliverables associated with that phase are accepted by the NCM. The next phase in the development cycle is not generally initiated until the NCM has accepted the deliverables associated with the current phase.

DELIVERABLES

Table 1 defines the deliverables required of the New Capability Contractor (NCC). Table 2 relates these deliverables to the phases of the new capability development cycle. Item numbers in Table 2 refer to item numbers in Table 1.

For most deliverables, both a preliminary version and a final version are required. The preliminary version is defined to be the NCC's best attempt at a final product. As indicated in Table 1, preliminary versions are to be reviewed by the NCM. The results of an NCM review are to consist of comments, required changes, and suggested changes as well as rejection or conditional acceptance of the deliverable. If a deliverable is rejected, the NCM is to inform the NCC of the reasons for rejection and what has to be accomplished prior to the resubmission of the preliminary version. If a deliverable is conditionally accepted, this implies the preliminary version is deemed sufficiently close to be an acceptable final product. After appropriate discussion and negotiation between the NCM and the NCC, the NCM's comments and agreed upon (negotiated) changes will be implemented by the NCC into the preliminary version. The final version is nothing more than the preliminary version modified to include the set of negotiated changes.

The NCC's deliverables with regard to the four NASTRAN manuals, The Theoretical Manual, The User's Manual, The Programmer's Manual, and The Demonstration Problem Manual, shall take one of two forms; new manual pages typed by the NCC on Government-furnished paper (mats) or completed Documentation Change Reports (DCRs). A DCR shall be used when a minor change to an existing manual page is required. Changes described on DCRs shall be incorporated by the NASTRAN Maintenance Contractor (MC) onto existing mats in the NCM's possession. A DCR is shown in Figure 1. DCRs shall be delivered as part of the preliminary version and final version of appropriate manuals and are therefore not shown in Tables 1 or 2.

Certain deliverables are classified as baseline documents. A baseline document is one which forms a foundation for subsequent deliverables. When information in a baseline document is added, deleted or changed, a formal written update to the baseline document is required. The change must be approved by the NCM before work can begin. In the new capability development cycle, there are two baseline documents: the Requirements and Mathematical Specification document and the Design Specification document. Because of the importance of these documents, each shall be reviewed in formal reviews. The Requirements and Mathematical Specification shall be reviewed in the Definition Phase Review, and the Design Specification shall be reviewed in the Design Phase Review. Details on these two reviews are given in Sections 2.2 and 2.3, respectively.

Table 1. Schedule of Deliverables.

Item No.	Description	Version	Section Reference	Delivery Date	No. Copies to the NCM	Activity
1	Requirements and Mathematical Specification	P	2.2	During Definition Phase	5	R
2	Requirements and Mathematical Specification	F	2.2	End of Definition Phase	5	A
3	Design Specification	P	2.3	During Design Phase	5	R
4	Test Plan and Test Specification	P	2.3	During Design Phase	5	R
5	Design Specification	F	2.3	End of Design Phase	5	A
6	Test Plan and Test Specification	F	2.4	End of Programming Phase	5	A
7	Theoretical Manual	P	2.4	End of Programming Phase	5	R
8	User's Manual	P	2.4	End of Programming Phase	5	R
9	Demonstration Problem Manual	P	2.4	End of Programming Phase	5	R
10	Source Listing Package	P	2.4	End of Programming Phase	1	R
11	Programmer's Manual	P	2.5	End of System Test Phase	5	R
12	Source Listing Package	F	2.5	End of System Test Phase	1	A
13	Test Results Report	P	2.5	End of System Test Phase	1	I,R
14	Tape and Card Package	F	2.5	End of System Test Phase	1	S
15	Theoretical Manual	F	2.6	End of Installation Phase	5	A
16	User's Manual	F	2.6	End of Installation Phase	5	A
17	Demonstration Problem Manual	F	2.6	End of Installation Phase	5	A
18	Programmer's Manual	F	2.7	During Acceptance Phase	5	A
19	Test Results Report	F	2.7	During Acceptance Phase	5	A

Legend:
A = Approval
F = Final Version
I = Information
NCM = NASTRAN Contract Monitor
P = Preliminary Version
R = Review
S = System Generation

Table 2. Documentation Deliverables Associated with the Phases of the New Capability Development Cycle..

New Capability Development Phase \ Document (Item Numbers)[1]	Requirements and Mathematical Specification (1,2)	Design Specification (3,5)	Test Plan and Test Specification (4,6)	Theoretical Manual (7,15)	User's Manual (8,16)	Programmer's Manual (11,18)	Demonstration Problem Manual (9,17)	Source Listing Package (10,12)	Test Results Report (13,19)	Tape and Card Package (14)
Definition Phase	P,F									
Design Phase		P,F	P							
Programming Phase			F	P	P		P	P		
System Test Phase						P		F	P	F
Installation Phase				F	F		F			
Acceptance Phase						F			F	

[1]Refer to Table 1

Legend: P = Preliminary Version
F = Final Version

DCR No. _____
(To be assigned by the NCM)

NASTRAN DOCUMENTATION CHANGE REPORT (DCR)

Organization: _____ Date: _____

Originator: _____ Phone No.: _____

Manual
- ☐ Theoretical _____
- ☐ User's _____
- ☐ Programmer's _____
- ☐ Demo. Problem _____

} Page Numbers

Reason for the Change: _____

Description of Change:
(Attach a copy of the page(s) to be changed with corrections typed. Use separate pages if necessary.)

Comments:

Figure 1. NASTRAN documentation change report (DCR).

2.2 DEFINITION PHASE

During the Definition Phase, the technical problem, stated in the RFP, is refined so that problems to be solved, problems not to be solved, requirements to be met, and the mathematical solution of the problem are specified by the NCC.

There is only one document associated with the Definition Phase, the Requirements and Mathematical Specification, which describes the requirements that the new capability must satisfy (i.e., the problem(s) to be solved by the enhanced NASTRAN system) and the mathematical solution to the problem. A sample outline for the Requirements and Mathematical Specification is given in Table 1. The NCM is to perform an indepth review of the Requirements and Mathematical Specification. After the NCM review of this specification, a formal Definition Phase Review shall take place. NCC attendees will include those personnel most familiar with the technical details so specific NCM questions may be answered.

The following points about the Requirements and Mathematical Specification shall be noted:

1. Section 2 contains a complete statement of the problem to be solved and the requirements that the NCC's product must meet. Included here are statements about user requirements, software requirements, reliability requirements, and documentation requirements.
2. Section 3 contains complete mathematical statements of the requirements and the solution to the requirements. It shall be written to agree as closely as possible with the style, level of detail, and content of The NASTRAN Theoretical Manual.
3. The Glossary shall be written with a view toward eventually incorporating its entries into the NASTRAN Dictionary, Section 7.1 of The NASTRAN User's Manual.

Table 1. Sample Outline for the Requirements and Mathematical Specification.

Section
1. INTRODUCTION
1.1 PURPOSE
1.2 SCOPE
2. REQUIREMENTS
3. MATHEMATICAL SPECIFICATION
GLOSSARY
REFERENCES

2.3 DESIGN PHASE

Once the NCM has accepted the Requirements and Mathematical Specification, the NCC shall examine candidate software solutions to the problem. After appropriate trade-off analyses, the NCC shall choose one software solution to the problem. The documentation of this solution is called the Design Specification, which, like the Requirements and Mathematical Specification, is a baseline document and is therefore subject to the change control procedures described later in Section 2.8. A sample outline of the Design Specification is given in Table 1.

The NCM is to perform an in-depth review of the Design Specification. After this review a formal Design Phase Review will take place. NCC attendees shall include those personnel most familiar with the technical details of the design so specific NCM questions may be answered.

Note that the two-digit section titles of Section 2 of the Design Specification parallel the section titles in The NASTRAN User's Manual and that the two-digit section titles of Section 3 of the Design Specification parallel the section titles in The NASTRAN Programmer's Manual. Conformance to the outline of Sections 2 and 3 is to ensure completeness of the Design Specification and easy transition of the deliverables to The NASTRAN User's Manual and The NASTRAN Programmer's Manual. The following points should be noted:

1. In Section 2.1, user-oriented modeling documentation shall be included in the form of updates to Section 1 of The NASTRAN User's Manual.

2. In Section 2.2, user-oriented modeling documentation shall be included in the form of updates to Section 2 of The NASTRAN User's Manual. All new Bulk Data cards shall be described in detail. The NCC's design shall be structured, whenever possible, to use existing Case Control cards as opposed to designing new ones.

3. New or modified Rigid Formats shall be included in Section 2.3. Solution subset definitions shall also be given. User-oriented requirements documentation shall be included in the form of updates to Section 3 of The NASTRAN User's Manual.

4. Particular attention shall be paid to diagnostic messages and explanations (see Section 4.2.3 of this document) contained in Section 2.4.

5. Section 2.5, OTHER USER REQUIREMENTS, refers to other sections of The NASTRAN User's Manual, for example, Plotting and DMAP.

6. The contents of Section 3.1 shall be an overview of the software design, but it is not intended to be a potential addition to Section 1 of The NASTRAN Programmer's Manual.

7. Section 3.2 shall contain documentation updates for Section 2 of The NASTRAN Programmer's Manual.

8. In Section 3.3, all new subprograms that are not an integral part of a module (i.e., those subprograms that shall be documented in Section 3 of The NASTRAN Programmer's Manual), shall be documented in the format given for prologue comments in Section 4.2.2 of this document. The use of actual prologue comments is encouraged. If there are any Assembly language routines in the design, the NCC shall have received NCM's permission for the use of such Assembly language (see Section 4.2.7 of this document for further details). Each Assembly language routine documented here shall include a functional flowchart and a detailed flowchart, which shall contain detailed logic showing all flow paths.

9. Section 3.4 shall contain preliminary module functional descriptions to the level of detail that exists in Section 4 of The NASTRAN Programmer's Manual. The module functional description for module ØPTPR1, Section 4.120 of The NASTRAN Programmer's Manual, shall be used as a model to illustrate the level of detail required. Each preliminary module functional description shall contain a module overlay diagram, an open core map, and a subprogram hierarchy map. Samples of each of these are given in Section 2.4, Figures 1, 2, and 3, respectively.

10. Section 3.5 shall contain updated link diagrams (see Section 5 of The NASTRAN Programmer's Manual) to indicate all control sections for all subprograms described in Sections 3.3 and 3.4.

11. Section 3.6 shall contain information on how to modify or add to the new capability software.

12. All support programs shall be documented in Section 3.7 to conform to the format of Section 7 of The NASTRAN Programmer's Manual.

13. All new elements will be documented in Section 3.8 to conform to the elements in Section 8 of The NASTRAN Programmer's Manual.

14. With NCM's permission given in a Special Purpose IRD, detailed mathematical algorithms

may be documented in Section 3.9 rather than in a module functional description.

15. Restart tables shall be documented in Section 3.10 to conform to the tables in Section 10 of The NASTRAN Programmer's Manual.

16. Appendixes might include descriptions of trade-off studies or analyses used to choose the proposed design.

17. The Glossary shall be written with a view toward eventually incorporating its entries into the NASTRAN Dictionary, Section 7.1 of The NASTRAN User's Manual.

The preliminary version of the Test Plan and Test Specification shall be delivered at the same time as the Design Specification; a sample outline is given in Table 2. The Test Plan and Test Specification identifies plans and specifications for two types of tests: integration and system. Integration tests, sometimes called package or subsystem tests, are performed to test lower-level interfaces. System tests are performed to provide confidence that previously existing capabilities and new capabilities are functioning properly. Three sets of system tests shall be performed by the NCC on two of the three versions of NASTRAN. See Section 2.5 of this document for a discussion of the three sets of system tests.

The following points about the outline for the Test Plan and Test Specification should be noted:

1. In Section 2.1, FEATURES TO BE TESTED, the NCC shall list all features to be tested (the running of system tests shall be included in the list of features to be tested). The correct execution of integration and system tests shall demonstrate that no major existing capability has been rendered inoperative. The tables, Features of NASTRAN to be Demonstrated, which can be found at the front of The NASTRAN Demonstration Problem Manual, can be used as a format model by the NCC to document features to be tested. It is expected, however, that the list of features to be tested shall contain more detail than shown in those tables, which were used for demonstration, not merely testing, of all the Rigid Formats.

2. Section 2.2, TEST COVERAGE MATRIX, will be modeled after the UMF Problem Numbers and Problem Restarts table at the front of The NASTRAN Demonstration Problem Manual.

3. For each integration and system test, the Overview in the test specification shall include test objectives, test philosophy, and the criteria for test success. Under Model and Data,

the mathematical model and finite-element model shall be described as shall the Executive Control, Case Control, and Bulk Data Decks. Checklists may include generation of part of the Bulk Data Deck by a utility program, key portions of the output, notes to use existing test problems or documentation, and so on. (See Section 2.5, for more details on system tests.)

Table 1. Sample Outline of the Design Specification.

Section	
1.	INTRODUCTION
	1.1 PURPOSE
	1.2 SCOPE
2.	USER-ORIENTED DESIGN IMPLICATIONS
	2.1 MODELING
	2.2 NASTRAN DATA DECK
	2.3 RIGID FORMATS
	2.4 DIAGNOSTIC MESSAGES
	2.5 OTHER USER REQUIREMENTS
3.	SOFTWARE-ORIENTED DESIGN IMPLICATIONS
	3.1 OVERVIEW
	3.2 DATA BLOCK AND TABLE DESCRIPTIONS
	3.3 SUBROUTINE DESCRIPTIONS
	3.4 MODULE FUNCTIONAL DESCRIPTIONS
	3.5 LINK DIAGRAMS
	3.6 MODIFICATIONS AND ADDITIONS
	3.7 SUPPORT PROGRAMS
	3.8 ELEMENT DESCRIPTIONS
	3.9 MATHEMATICAL ALGORITHM DESCRIPTIONS
	3.10 RESTART TABLES
APPENDIXES	
GLOSSARY	
REFERENCES	

Table 2. Sample Outline of the Test Plan and Test Specification

Section
1. INTRODUCTION
1.1 PURPOSE
1.2 SCOPE
2. TEST PLAN
2.1 FEATURES TO BE TESTED
2.2 TEST COVERAGE MATRIX
3. TEST SPECIFICATIONS
3.i SPECIFICATIONS FOR TEST i
3.i.1 Overview
3.i.2 Model and Data
3.i.3 Checklists
REFERENCES

2.4 PROGRAMMING PHASE

In the Programming Phase, the software solution is implemented and partially tested, i.e., integration testa are performed, and a great deal of preliminary documentation is delivered. The Programming Phase ends with delivery of preliminary versions of the following items:

1. Updates to The NASTRAN Theoretical Manual
2. Updates to The NASTRAN User's Manual
3. Updates to The NASTRAN Demonstration Problem Manual
4. Source Listing Package (See Table 1)
5. Test Plan and Test Specification

The parts of the Source Listing Package are listed in Table 1. The number 10 and the letter L refer to the 10th Deliverable Item in Section 2.1, Table 1 and the word listing, respectively. Figure 4 is an example of a Module Subprogram Cross-Reference Table (10L5). Figure 5 is an example of a Module Subprogram CØMMØN Block Cross-Reference Table (10L6). Figures 6 and 7 are examples of the NASTRAN NCC Alter Form.

Table 1. Itemized Deliverables in the Source Listing Package.

Item Number[1]	Description
10L1.	Compiler Listings and Compiler Maps for New Subprograms
10L2.	Compiler Listings and Compiler Maps of Modified Subprograms After Alters Have Been Made
10L3.	Listing of the SUBSYS Deck After Alters Have Been Made (Two Machines)
10L4.	Load Maps (Linkage Editor Output) for Two Machines
10L5.	Module Subprogram Cross-Reference Table
10L6.	Module subprogram CØMMØN Block Cross-Reference Table

[1]Refer to Section 2.1, Table 1.

Legend: L = Listing

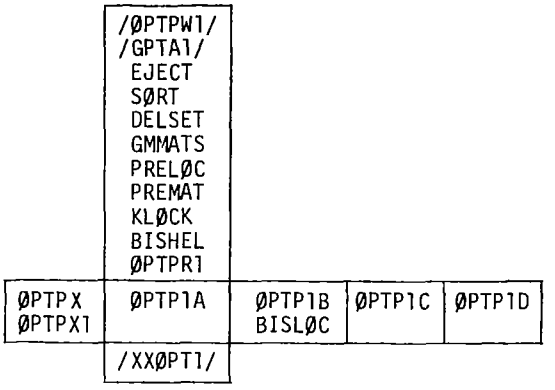


Figure 1. Sample module (ØPTPR1) overlay diagram.

X(1)	PØPT(6) - Data from the PØPT bulk data card
Y(1)	Material Data
Y(PCØR1)	ELT(NTYPES) - Element type pointers in /GPTA1/ sequence. Contains zeros for elements to skip and an increasing integer sequence for types to optimize (NPØW types are non-zero).
Y(PCØR2)	ELØP(2,NPØW+1) - Element type and element property starting locations in ELE and PR arrays. ELØP(1,I) for elements and ELØP(2,I) for properties where I is the ELT value. The difference between succeeding entries is the number of words used for the element type.
Y(ECØR1)	ELE(NELW) - Element section of core where NELW is the number of words. Each entry is as follows: ELE(1) - Element ID ELE(2) - σ_t ELE(3) - σ_c } Stress limits from material cards ELE(4) - σ_s ELE(5) - Property card pointer relation to ELØP(2,1)
Y(PRØCR1)	PR(NPRW) - Property section of core where NPRW is the number of words. Each entry is as follows: PR(1) - Property ID PR(2) - EST word and element stress limit to optimize PR(3) - Original property PR(4) - Last property value - initially PR(3) = PR(4) PR(5) - Property change ratio PR(6) - Property change limit pointer relative to PL(1)
Y(KCØR1)	PL(NKLW) - Property change limits where NKLW is the number of words. Each entry is as follows: PL(1) - minimum property $\left(\frac{\text{NEW}}{\text{ØRIGINAL}}\right)$ PL(2) - maximum property $\left(\frac{\text{NEW}}{\text{ØRIGINAL}}\right)$

Figure 2. Sample open-core map (ØPTPR1).

ØPTPR1

(Fully Stressed Design Phase 1) Link 2

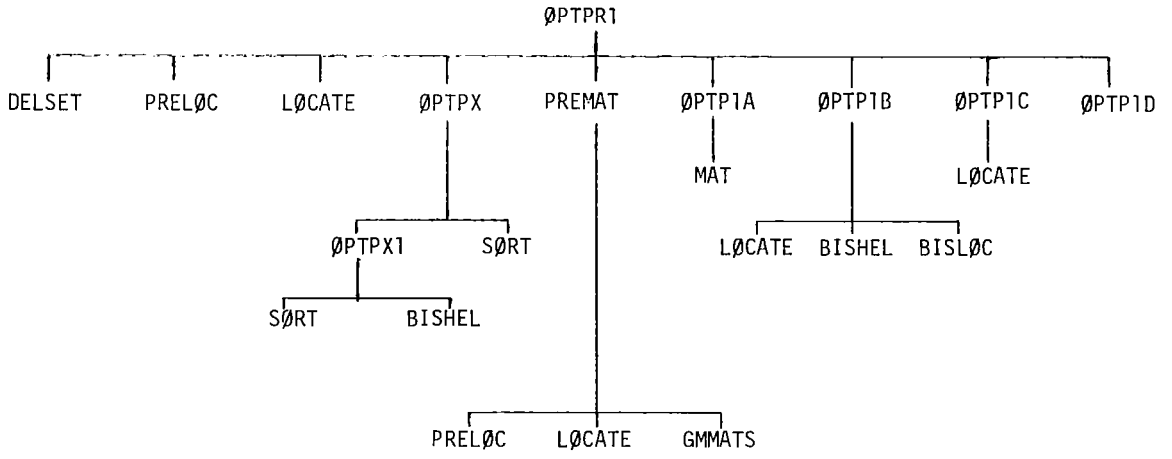


Figure 3. Sample subprogram (ØPTPR1) hierarchy map.

Calling Sub-Program	Called Subprogram															
	ØPTPR1	ØPTPX	ØPTPX1	ØPTP1A	ØPTP1B	ØPTP1C	ØPTP1D	PRELØC	LØCATE	SØRT	BISHEL	PREMAT	GMMATS	DELSET	BISLØC	MAT
ØPTPR1		X		X	X	X	X	X	X			X		X		
ØPTPX			X							X						
ØPTPX1										X	X					
ØPTP1A																X
ØPTP1B									X		X				X	
ØPTP1C									X							
ØPTP1D																
PRELØC																
LØCATE																
SØRT																
BISHEL																
PREMAT								X	X				X			
GMMATS																
DELSET																
BISLØC																
MAT																

Figure 4. Module subprogram cross-reference table.

Calling Subprogram \ CØMMØN Block	ØPTPW1	NAMES	GPTA1	BLANK CØMMØN	XXØPT1	SYSTEM	MATØUT	MATIN	TWØ	SETUP
ØPTPR1	X	X	X	X	X	X				
ØPTPX	X	X	X	X	X	X				
ØPTPX1				X	X	X				
ØPTP1A	X	X	X	X		X	X	X		
ØPTP1B	X	X	X	X	X	X				
ØPTP1C	X	X	X	X	X	X				
ØPTP1D	X	X		X		X				
PRELØC				X					X	
LØCATE				X					X	
SØRT						X				X
BISHEL				X						
PREMAT		X		X		X	X	X		
GMMATS				X						
DELSET			X	X		X				
BISLØC				X						
MAT		X				X	X	X		

Figure 5. Module subprogram-CØMMØN block cross-reference table.

NEW CAPABILITY: Rigid Format Improvements	DATE: 7/4/76	PAGE: 1 OF 1
SUBROUTINE: LD12	LEVEL ALTERED: 15.1	[X] CDC [X] IBM [X] UNIVAC

PURPOSE OF ALTER: To allow checkpointing in Rigid Format 12, Modal Transient Response.

ALTER: List actual code used to produce the alter together with system control cards needed to add to or delete from existing code. Explain reason for the alter if not apparent by including comment cards in the alter.

Because the CDC box is checked above, the following alter would be reported:

```
*DELETE,LD12.110
    bbbbb*bb,4HPURG,4HEbGM,4H,GMD,4H/MPC,4HF1/G,4HØ,GØ,4HD/ØM,4HIT/K,4HFS,Pbbbbbbbbb
*CØMPILE LD12
This line avoids a fatal DMAP compiler error message.
```

Because the IBM box is checked above, the following alter would be reported:

```
./ CHANGE NAME=LD12
    bbbbb*bb,4HPURG,4HEbGM,4H,GMD,4H/MPC,4HF1/G,4HØ,GØ,4HD/ØM,4HIT/K,4HFS,Pb00001090
This line avoids a fatal DMAP compiler error message.
```

Because the UNIVAC box is checked above, the following alter would be reported:

```
@FØR,SW   SØU.LD12,ØBJ.LD12,ØBJ.LD12
-109,109
    bbbbb*bb,4HPURG,4HEbGM,4H,GMD,4H/MPC,4HF1/G,4HØ,GØ,4HD/ØM,4HIT/K,4HFS,Pbbbbbbbbb
This line avoids a fatal DMAP compiler error message.
```

Figure 6. Sample NASTRAN NCC alter report for a coding change.

NEW CAPABILITY: Dummy Element Matrix Assembly	DATE: 7/4/76	PAGE: 1 OF 1
SUBROUTINE: LINK8	LEVEL ALTERED: 16.0	[X] CDC [ ] IBM [ ] UNIVAC
PURPOSE OF ALTER: To convert the element matrix assembly for DUMMY elements to EMG technique.		

ALTER: List of actual code used to produce the alter together with system control cards needed to add to or delete from existing code. Explain reason for the alter if not apparent by including comment cards in the alter.

Because the CDC box is checked above, the following alter would be reported:

```
*DELETE LINK8.192,LINK8.209
```

This removes the Overlay structure for the DUMMY elements' individual structural matrix assemblies.

```
*INSERT LINK8.309

ØVERLAY EMGELS

INCLUDE NASTØBJ(KDUM1,KDUM2,KDUM3,KDUM4,KDUM5,KDUM6,KDUM7,KDUM8,KDUM9)

*CØMPILE LINK8
```

This alter incorporates the EMG technique for the DUMMY elements.

Figure 7. Sample NASTRAN NCC alter report for a SUBSYS change.

2.5 SYSTEM TEST PHASE

After the NCC has successfully completed integration tests, system tests (divided into three subsets) shall be performed on two of the three NASTRAN machines to demonstrate that previously developed NASTRAN capabilities still work correctly and that newly implemented software correctly solves the problem stated in the Requirements and Mathematical Specification.

The demonstration that previously developed NASTRAN capabilities still work correctly shall be in the form of the successful execution of Subset 1 and Subset 2 of problems from the NCM's test library. Subset 1 will be provided to all NCCs and is described in Table 1. This is a set of selected demonstration problems, documented in The NASTRAN Demonstration Problem Manual, the results of which must be reproducible before and after NCC delivery. Subset 2 shall be supplied specifically for each NCC and will be specified in a Special Purpose IRD. This is a set of specific problems emphasizing the area in which the NCC is working. Tests to show that the new capability software functions in conformance with the Requirements and Mathematical Specification make up Subset 3, which shall be specified by the NCC in Section 3 of the Test Plan and Test Specification.

At the end of the System Test Phase, the following items will be delivered to the NCM:

1. Updates to The NASTRAN Programmer's Manual (preliminary version)
2. Source Listing Package (final version)
3. Test Results Report (preliminary version)
4. Tape and Card Package (final version)

2.5.1 Test Results Report

A sample outline of the Test Results Report is given in Table 2. Significant features included in the report are as follows:

1. The scope of the Test Results Report shall include integration and system tests.
2. Section 2 shall be written to stand alone and should be brief.
3. Section 2.2, TEST RESULTS MATRIX, shall have a format similar to the Test Coverage Matrix of the Test Plan and Test Specification. The matrix entries shall indicate the success or failure of the test.

4. For each failure entered in the Test Results Matrix, a discussion shall be included in Section 2.3. Each discussion shall include the test number; identification of the modules or subprograms in which the problem occurred (if known); a description of the problem, with supporting data (if available); and recommendations for solutions to the problem.

5. For each test conducted in the System Test Phase, the computer output shall be given in Section 3. Output shall be clearly marked so that test results can be easily keyed to the Test Coverage Matrix. For integration tests, results on one computer are necessary; for system tests, results on two computers are necessary.

6. Section 4 shall not be part of the preliminary version of the Test Results Report. Section 4 shall contain results of all tests conducted during the Installation Phase and the Acceptance Phase. All system tests, Subsets 1, 2, and 3, shall be executed in the Installation Phase. All NCM-developed acceptance tests shall be executed in the Acceptance Phase.

7. For the final version of this report, Section 2 shall be updated to reflect the results of tests conducted during the Installation and Acceptance Phases.

The final, and only, version of the Tape and Card Package, Deliverable Item 14, is delivered at the end of the System Test Phase. The components of this package are listed in Section 2.7, Table 1. Items listed 14Ti shall be delivered on tape; items listed 14Cj shall be delivered on FØRTRAN cards. Source cards shall be delivered only in BCD format for compilation on any of the compilers used to support NASTRAN. The format and file structure for tape deliverables shall be specified in a Special Purpose IRD.

Table 1. Subset 1 of System Test Problems.

Problem Type	Problem Number	Rigid Format	Test Problem
Displacement	Demo 1-1-1	1	Static Analysis of a Delta Wing with Checkpoint
Displacement	Demo 1-1-1B	3	Restart of Demo 1-1-1 for Normal Modes Analysis (Rigid Format Switch)
Displacement	Demo 5-1-1	5	Buckling Analysis of a Flat Plate with Structure Plots
Displacement	Demo 8-1-3	8	Frequency Response Analysis of a Flat Plate via INPUT Module
Displacement	Demo 9-3-1	9	Transient Analysis of a Fluid-Filled Elastic Cylinder with X-Y Plots
Displacement	Demo 12-1-1	12	Transient Response of a Beam with Structure and X-Y Plots
Displacement	Demo 15-1-1	15	Normal Modes Analysis with Cyclic Symmetry
Heat	Demo 1-12-2	1	Linear Steady State Heat Conduction using Solids of Revolution
Heat	Demo 3-6-1	3	Radiation Heat Transfer in a Sphere
Heat	Demo 9-4-1	9	Transient Heat Transfer Analysis of a Flat Plate
Aero	Demo 10-2-1	10	Aeroelastic Flutter Analysis of a Swept Wing

Table 2. Sample Outline for the Test Results Report.

Section
1. INTRODUCTION
1.1 PURPOSE
1.2 SCOPE
2. SUMMARY OF TEST RESULTS
2.1 INTRODUCTION
2.2 TEST RESULTS MATRIX
2.3 PROBLEMS ENCOUNTERED
3. TEST RESULTS DURING THE SYSTEM TEST PHASE
3.i RESULTS FOR TEST i
4. TEST RESULTS DURING THE INSTALLATION PHASE AND ACCEPTANCE PHASE
4.j RESULTS FOR TEST j

2.6 INSTALLATION PHASE

The new capability software developed by the NCC shall be integrated by the Maintenance Contractor (MC) into the current in-house NASTRAN level on the primary designated computer and on one of the other two computers used to support NASTRAN. The new capability shall be certified by the NCM through its MC. The integration process shall take into account the fact that the source library the NCC used as a baseline to form the test-bed system is, in general, different from the more up-to-date source library of the current in-house level of NASTRAN. If the NCM deems it necessary, NCC personnel shall be available, at the MC's on-site facility, to work with MC personnel to ensure that any problems with the integration and installation of the new capability software shall be addressed by the personnel who developed the new code.

The deliverables (final version) associated with the Installation Phase are:

1. Updates to The NASTRAN Theoretical Manual
2. Updates to The NASTRAN User's Manual
3. Updates to The NASTRAN Demonstration Problem Manual

2.7 ACCEPTANCE PHASE

Initial acceptance of the NCC's product shall be based upon receipt of the articles described in Section 2.4, Table 1 and in Table 1 of this section; the listings of successful executions of the system test problems; and successful executions of the acceptance tests on the primary designated computer. Final acceptance of the NCC's product shall be based upon successful execution of all acceptance tests once the capability is installed in the in-house level of NASTRAN.

The deliverables (final version) associated with the Acceptance Phase are:

1. Updates to The NASTRAN Programmer's Manual
2. Test Results Report

Table 1. Itemized Deliverables in the Tape and Card Package.

Item Number[1]	Description
14T1	Source for New Subprograms
14T2	Source for Modified Subprograms After Alters Have Been Made
14T3	SUBSYS File After Alters Have Been Made (Two Machines)
14T4	Executable Including New Capability
14T5	Object Modules for All New Subprograms and Modified Subprograms
14C1	Alters Necessary to Generate Modified Subprograms
14C2	Alters Necessary to Generate Modified SUBSYS (Two Machines)
14C3	Card Decks for All NCC-developed New Capability System Test Problems

[1]Refer to Section 2.1, Table 1.

Legend: T = Tape
C = Cards

2.8 CHANGE CONTROL PROCEDURES

Because of the importance of up-to-date information in the Requirements and Mathematical Specification and the Design Specification, these two documents are designated baseline documents, which implies that both shall be formally updated. The NCM neither expects nor desires any elaborate changes. Requested changes to these two basic documents must be forwarded to the NCM in a clearly marked fashion, as soon as they are needed. The NCM is to then review and either approve or disapprove the requested changes. The fact that a separate section of this document is devoted to change-control procedures shows the NCM's recognition that change is inevitable in the software development cycle. This recognition should not, however, be construed as a license for incomplete specifications. The NCM shall require a formal Definition Phase Review at which the Requirements and Mathematical Specification shall be reviewed and a formal Design Phase Review at which the Design Specification shall be reviewed. Because of the investment associated with these specifications and reviews, it is expected that the NCC shall have taken all the steps necessary for completeness of these specifications.

As stated in Section 2.1, a preliminary version is defined to be the NCC's best attempt at a final product, and the corresponding final version is the preliminary version modified to include the set of negotiated changes. The negotiated changes shall generally be fairly minor. The time for preparation of the final version shall not be construed as a time for adding major capability to the preliminary version of the document.

3. ADDING TO AND MODIFYING NASTRAN DOCUMENTATION

3.1 DOCUMENTATION OVERVIEW

There are four official NASTRAN manuals that a New Capability Contractor (NCC) must add to or modify (using the latest public edition): The NASTRAN Theoretical Manual, NASA SP-221; The NASTRAN User's Manual, NASA SP-222; The NASTRAN Programmer's Manual, NASA SP-223; and The NASTRAN Demonstration Problem Manual, NASA SP-224. An overview of the manuals is presented in the introduction to each one, and the NCC is expected to read them and have them ready for reference during the task of adding to and modifying the manuals.

For most NASTRAN users and programmers, the NASTRAN manuals are used basically for reference. NCC personnel, however, must have a greater familiarity and facility with them. This is true not only in designing and implementing the new capability but in adding to or modifying the NASTRAN manuals.

To help ensure that all documentation requirements are met and interfaces properly defined, it is strongly urged that the sections of the Design Specification (see Section 2.3) be documented, to the greatest extent possible, in the form of additions and modifications to the NASTRAN manuals.

3.2 FORMAT RULES

Section 3.2.1 contains format specifications for NASTRAN manuals that were formulated during the original NASTRAN development effort and augmented using experience gained during maintenance efforts. Section 3.2.2 defines standards for flowchart symbols.

3.2.1 NASTRAN Documentation Format Specifications

This section contains specifications to be followed to ensure conformance to the current documentation format. A good, consistent format cannot compensate for bad content, but bad format can distract from effectively communicating content that is good; therefore, conformance with these specifications by writers and typists is very important. Most documentation specialists and typists look to the current documentation and use it as a model. It is expected that this approach shall continue to be applied. These specifications are intended to complement (not supersede) this approach. The NCC's good judgment will decide conflicts. The Maintenance Contractor (MC) is available, through the NASTRAN Contract Monitor (NCM), to help in decision making, if necessary.

The specifications below are divided into eight general categories:

1. General Information
2. Spacing
3. Table of Contents
4. Section Numbering
5. Page Starting, Page Numbering, and Running Headings
6. Equations
7. Tables, Figures, and References
8. Capitalization
9. Punctuation

It is expected that the NCC's professional and support staffs shall follow the specifications to ensure mutual understanding before beginning the writing or typing of NASTRAN documentation. Effective documentation cannot be accomplished without a thorough familiarity of the style and content of each manual and the required cross-referencing among them.

3.2.1.1 General Information

1. NASTRAN documentation must be typed on 11- by 14-inch mats that will be supplied by the NCM. (The mats will be reduced to 80 percent of their original size during the printing process. Typists should take this into consideration when they use printed documentation as a model for document updating.) There are two exceptions to this rule. First, the title page of each manual is typed on 8-1/2- by 11-inch mats, which are printed at full size type. Second, the link maps in Section 5 of The NASTRAN Programmer's Manual are taped or pasted to 11- by 14-inch mats, which are themselves taped side by side to form a continuous n- by 14-inch oversize mat for each link. These mats will be printed at an 80-percent reduction (continuous sheet) and will be folded by the printer to fit the manual.

2. Text will be typed on a 12-pitch IBM Selectric using Letter Gothic, ball number 005. Carbon ribbon will be used.

3. Mathematical symbols and Greek characters will be typed on a 12-pitch IBM Selectric using Symbol 10, ball number 061 or Symbol 12, ball number 004.

4. It is imperative that computer printout used in the manuals (for example, see the Rigid Formats in Section 3 of The NASTRAN User's Manual) be clear and sharp. To ensure this. the reverse side of the computer paper should be used, and a new ribbon should be inserted on the printer. For printing a new Rigid Format (including headings), use LINE=52, where LINE is a card in the Case Control Deck. A complete listing of a Rigid Format includes a TITLE, SUBTITLE, and LABEL from Case Control cards for that purpose (see Section 3 in The NASTRAN User's Manual).

5. To change camera-ready manuscripts, correction tape is preferred to mortising (cutting and pasting). Also, for one-letter corrections, a chalk-like substance may be used. Erasures and opaque white correction fluid are not acceptable.

6. Minor modifications to pages of existing NASTRAN documentation shall be communicated to the NCM by means of the Documentation Change Request (DCR), which is described in Section 2.1 and shown in Figure 1 of that section.

3.2.1.2 Spacing

1. Double spacing shall be used except where groups of a few single spaced lines separated by double spacing for the groups is more desirable for clarity or appearance. Examples of the exceptions include Section 2.3, DATA BLOCK DESCRIPTIONS, of The NASTRAN Programmer's Manual and Section 2.4, BULK DATA DECK, of The NASTRAN User's Manual.

2. Paragraphs shall be indented five spaces and shall be separated from each other by two and one-half spaces. Listed information shall be indented five spaces, i.e., the numbers 1 through 9 associated with the listing shall start in the sixth space and shall be followed by a period, and the text shall start in the tenth space. Numbers 10 and beyond shall be backspaced one space to preserve the alignment of the period. Continuation lines shall start in the tenth space. Each numbered item shall be treated as a paragraph for the purposes of line spacing.

3. Section titles shall be separated from the text (above and below the title) and from each other by three lines.

3.2.1.3 Table of Contents

The Table of Contents shall list all section titles and the page number on which they begin. One- and two-digit section titles shall appear in all capitalized letters. Section titles of three digits and beyond shall have only the initial letters of each word capitalized. The word Section (underlined) shall appear in the upper left corner of the mat text boundary and Page No. (underlined) shall appear in the upper right. The single-digit section number shall begin in the third space. Succeeding indentations are as described in the previous section. Double spacing shall be used except where long titles necessitate two lines, in which case single spacing shall be employed. A string of periods shall be used to fill the space from the last letter of the section title to two spaces prior to the first digit of the page number.

3.2.1.4 Section Numbering

1. One-digit section titles shall be typed in all capital letters, shall be centered at the top of the first page of the section, and shall be identified with a decimal classification and one number followed by a period, as follows:

10. EIGENVALUE EXTRACTION METHODS

2. Two-digit section titles shall be typed in all capital letters and shall be identified with a decimal classification and two numbers as follows:

10.4 THE INVERSE POWER METHOD WITH SHIFTS

3. Three-digit section titles shall be typed in initial capitals, shall be underlined, and shall be identified with a decimal classification and three numbers as follows:

10.4.2 Theory for Real Eigenvalue Analysis

4. Four-digit or greater section titles shall be typed in initial capitals, shall not be underlined, and shall be identified with a decimal classification and four or more numbers (four numbers being the preferred limit) as follows:

10.4.2.4 Sweeping Previously Found Eigenvalues

3.2.1.5 Page Starting, Page Numbering, and Running Headings

1. Two-digit sections shall begin at the top of an odd-numbered page (when printed, odd-numbered pages are on the right and even-numbered pages are on the left). Bulk Data card descriptions in Section 2.4 of The NASTRAN User's Manual and module descriptions in Section 4 of The NASTRAN Programmer's Manual shall also begin at the top of an odd-numbered page. Other units such as subprogram descriptions in Section 3 of The NASTRAN Programmer's Manual should begin at the top of a page when clarity or convenience is thereby improved. Within voluminous two-digit sections, three-digit sections may begin at the top of the next page. An example of the latter case is Section 2.3 of The NASTRAN Programmer's Manual.

2. Numbers for new pages shall be centered at the bottom of each page, shall be identified by the two-digit section identification number, a dash number, and shall be followed by a date in the form (mm/dd/yy) as follows:

2.1-1 (4/10/75)
2.1-2 (4/10/75)
2.1-3 (4/10/75)

The date to be used shall be prescribed in a Special Purpose IRD or other appropriate direction from the NCM.

3. Pages inserted at a later date between pages shall be identified by the two-digit section identification number, a dash number, a letter, and shall be followed by the date of insertion in the form (mm/dd/yy), as follows:

2.1-2	Original page
2.1-2a (4/10/75)	Added page
2.1-2b (4/10/75)	Added page
2.1-3 (3/1/72)	Previously changed page

If an odd number of pages is to be inserted, one blank page with a running heading and a page number should be added to ensure proper printing. A blank page shall contain the following sentence, centered in the middle of the page: THIS PAGE HAS BEEN LEFT BLANK INTENTIONALLY. If more than 26 pages are to be inserted, multiple letters shall be used after the 26th inserted page, as follows: za, zb, ... , zz, zaa, zab, ... , zaz, zaaa, zaab, etc.

4. Pages inserted at a later date between pages of insertions made subsequent to the original issue shall be identified by the two-digit section identification number, a dash number, two letters, and shall be followed by the date of insertion in the form (mm/dd/yy), as follows:

2.1-2	Original page
.	.
.	.
.	.
2.1-2c (9/1/71)	Previously added page
2.1.2ca (4/10/75)	Added page
2.1-2d (9/1/71)	Previously added page
.	.
.	.
.	.
2.1-3	Original page

5. Pages changed after the original issue shall be identified as above, as follows:

3.1-2	Original page
3.1-3 (4/10/75)	Changed page
3.1-4	Original page

6. Text added to an existing page shall be indicated by a vertical line in the outer margin except where more than half the page has been changed. Assuming this item were a change, the margin would be marked as shown.

7. Running headings, in capitals, shall be centered at the top of each page. The one-digit section name shall be used as the running heading on even-numbered pages, and the two-digit section name shall be used as the running heading on odd-numbered pages. There is one exception to this rule: for the first page in every two-digit section, the one-digit section name shall be used as the running heading. Care should be exercised in determining running headings for pages to be inserted. For example, the page to be inserted between 3.1-13 and 3.1-14 is 3.1-13a and is treated as an even-numbered page because it is printed on the back of 3.1-13 (an odd-numbered page). (A blank odd-numbered page, 3.1-13b, with running heading and page number should be typed to ensure proper printing. This blank page shall contain the following sentence, centered in the middle of the page: THIS PAGE HAS BEEN LEFT BLANK INTENTIONALLY.)

8. If pages (or whole sections) are deleted, a blank page is needed to fill the void and preclude the necessity of renumbering succeeding pages. When the deletion of a single page is required, a blank page with running heading and page number shall be handled as above. When the deletion of several pages in succession or a complete section is required, two blank pages with running headings and page numbers (for front and back) shall be used to fill the void. A sentence shall be centered in the middle of each page as appropriate: MATERIAL PREVIOUSLY ON PAGES a.b-c THROUGH x.y-z HAS BEEN DELETED or MATERIAL PREVIOUSLY IN THIS SECTION HAS BEEN DELETED. Never refer to specific NASTRAN levels, such as FOR LEVEL 15, or other time-related qualifiers.

9. A section may be assigned but left blank for a period of time, at the direction of the NCM, in order to logically fit information from different NCCs when final delivery is accomplished. In this case, two blank pages, as discussed above, shall contain the following sentence: THIS SECTION IS RESERVED FOR FUTURE USE.

3.2.1.6 Equations

1. Equations shall be numbered consecutively beginning with 1 for the first equation in each two-digit section. References to equations outside a section must refer to both the equation number and the section number, e.g., See Section 3.5, Equation 31. If the reference is to another manual, the manual name shall be given, e.g., See the Theoretical Manual, Section 3.5, Equation 41.

2. Equations shall be centered on the line and separated from the text both above and below by three blank lines. Equations shall be punctuated as part of the text and shall be identified in the right-hand margin with an Arabic numeral equation number in parentheses. Punctuation should not be counted when centering an equation. Punctuation shall be separated from the last symbol in the equation by five spaces to avoid conflicting punctuation periods with equation decimals. Equation development is shown in the following example where equations are a part of a sentence.

   The displacements in the x, y, and z directions are computed by

$$[K]\{U_x\} = \{P_x\} \quad , \tag{1}$$

$$[K]\{U_y\} = \{P_y\} \quad , \tag{2}$$

   and

$$[K]\{U_z\} = \{P_z\} \quad . \tag{3}$$

   However, the same equations may be developed when stated in a listed format as in the next example.

   The displacements in the x, y, and z directions are computed by the following equations:

$$[K]\{U_x\} = \{P_x\} \quad , \tag{4}$$

$$[K]\{U_y\} = \{P_y\} \quad , \tag{5}$$

   and

$$[K]\{U_z\} = \{P_z\} \quad . \tag{6}$$

3. When a group of equations appears in succession without text between them, the longest equation shall be centered and the equal signs of the remaining equations shall be aligned

with the equal sign of the longest equation. For example:

$$[K_{ii}] = [Z]^{-1} \quad , \tag{7}$$

$$[K_{id}] = -[Z]^{-1}[S] \quad , \tag{8}$$

and

$$[K_{dd}] = [S]^{T}[Z]^{-1}[S] \quad . \tag{9}$$

4. The transpose operator for a matrix should be placed outside the brackets (i.e., $[A]^{T}$ is correct; $[A^{T}]$ is incorrect). The same rule applies to the inverse operator (i.e., $[A]^{-1}$ is correct; $[A^{-1}]$ is incorrect).

5. All subscripts for matrices shall be lowercase letters (e.g., $[K_{gg}]$, $[K_{fs}]$).

6. All plus and minus signs in equations shall be preceded and followed by one space. There shall be two spaces before and after all equal signs in equations. There shall be no spaces between parenthetical expressions. For example:

$$[A][B] + [D] = [C] \quad . \tag{10}$$

7. Do not use the Symbol 10 ball for the Greek letter sigma (Σ); instead, use prestypes ($\mathbf{\Sigma}$) that are larger than the typed symbols and can accommodate the limits.

8. Equations inserted at a later date shall be designated by a number followed by a lowercase letter. For example, two equations inserted between Equation 5 and Equation 6 shall be designated by Equation 5a and Equation 5b, respectively. For more complicated cases, follow the rules given in items 3, 4, and 5 in Section 3.2.1.5 of this document.

3.2.1.7 Tables, Figures, and References

1. Tables and figures shall be numbered consecutively beginning with 1 for the first table or figure in each two-digit section. References to tables and figures outside a section must refer to both the table or figure number and the section number, e.g., See Section 3.4, Table 2. If the reference is to another manual, the manual name shall be given, e.g., See the User's Manual, Section 3.4, Table 2.

2. Table titles shall be typed with initial capitals centered above the table. Periods shall follow the Arabic table number and the end of the complete title as follows:

Table 3. Example of a Table Title.

The first line(s) of multiple-line table titles will be left-justified and the last line centered.

3. Figure captions shall be typed in lowercase letters except for the first letter of the first word. Periods shall follow the Arabic figure number and the end of the complete caption as follows:

Figure 4. Example of a figure caption.

Single-line captions shall be centered under the figure. Multiple-line captions shall be left-justified with the last line centered.

4. References shall be listed at the end of a two-digit section and shall be numbered consecutively beginning with 1 for the first reference.

5. Tables, figures, and references inserted at a later date shall be designated by a number followed by a lowercase letter. For example, two figures inserted between Figure 2 and Figure 3 shall be designated Figure 2a and Figure 2b.

6. The words equation, figure, reference, section, and table shall be spelled out with initial capitals when used with a number either in the text or in a caption. The associated Arabic numeral shall <u>not</u> be enclosed in parentheses.

7. References, tables, and figures are placed in that order at the end of a two-digit section.

3.2.1.8 Capitalization

1. Data block names, module names, Bulk Data card names, entry point names, and FØRTRAN variable names shall be in all capital letters with the letter O slashed (Ø). The following frequently used acronyms and terms are always capitalized:

DMAP, FIAT, FIST, FØRTRAN, GINØ, NASTRAN, NPTP, ØPTP, ØSCAR, PØØL, SØRT1, SØRT2

2. The following words shall have initial capitals whenever they appear:

Bulk Data Deck, Case Control Deck, Data Pool File, Executive Control Deck, Executive System, New Problem Tape, Old Problem Tape, Piecewise Linear Analysis, Preface, Problem Tape, Rigid Format.

3.2.1.9 Punctuation

1. Commas shall be used to separate the elements in a series. It is recommended that a comma be placed before the final conjunction. (Currently, some inconsistency exists in the manuals in this regard.)

This:	Modules IFP, XSØRT, and IFP1 are Preface modules.
Rather Than This:	Modules IFP, XSØRT and IFP1 are Preface modules.

2. In DMAP statements, do not leave a space before and after a slash.

```
This:              GPSP    GPL,GPST,USET,SIL/ØGPST/V,N,NØGPS $
Rather Than This:  GPSP    GPL,GPST,USET,SIL / ØGPST / V,N,NØGPS $
```

3.2.2 Flowchart Standards

In general, detailed flowcharts are not required, but the use of macro or functional flowcharts is encouraged. Flowchart symbols shall conform to the American National Standards Institute (ANSI) standard X3.5, which describes three groups of flowchart symbols: basic, additional, and specialized. The basic and the additional symbols are shown in Figure 1. Figure 2 shows specialized symbols for media, equipment, and processing. Complete flowcharts can be drawn using only the basic and the additional symbols. While the standard says the use of specialized symbols is optional, flowcharts drawn for NASTRAN shall use, where appropriate, all the specialized symbols for media and equipment and the decision, preparation, and predefined-process symbols, which are a subset of the specialized symbols.

The standard specifies the shape, but not the size, for the symbols in each group, thus permitting symbols of any size to be drawn. The shape is specified in two ways: by the ratio of the width to the height and by the general geometric configuration.

The standard specifies the use of a single width or weight of line for drawing the symbols. A single orientation or positioning of the symbols with respect to each other is assumed. In particular, those symbols shown horizontally oriented in the standard should be drawn that way.

3.2.2.1 Basic Symbols

The basic symbols are the input/output, processing, flowline, and annotation (see Figure 1(a)). The input/output symbol indicates an input or output operation, or input or output data. It is defined for use independent of media, format, equipment, and timing. The processing symbol is the general purpose symbol. It is the default symbol for use when no other symbol is specified. The processing symbol indicates data transformation, data movement, and logic operations.

The flowline symbol is an arrow of any length that connects successive other symbols to indicate the sequence of operations or data (the direction of flow). It is defined for use in an alternating fashion with other symbols. As such, it also indicates the sequence in which the other symbols are to be read. To specify the direction of flow or reading, arrowheads may be used on any flowline as shown in Figure 1(a).

The normal direction of flow is primarily from top to bottom and secondarily from left to right. Where the flow follows this normal pattern, no arrowheads are needed to remind the reader. In the event of any significant deviation from this pattern, arrowheads are required. Whenever the direction of flow might be ambiguous to a reader, arrowheads should be used to provide clarification.

The annotation symbol provides a way to supply description information, comments, and explanatory notes. Its dashed line indicates the symbols (including flowline symbols) to which this explanation or clarification applies.

3.2.2.2 Additional Symbols

The additional symbols, shown in Figure 1(b), are for the convenience of the reader, not for the purpose of describing data processing action. These symbols provide for handling the limitations of pages of various sizes and make it more convenient to show connections in the sequences of flow.

There are two varieties of connector symbols: an inconnector or entry connector and an outconnector or exit connector. Each inconnector may have any number of outconnectors associated with it. One function of the connector symbol is to enable a long sequence of symbols to be broken into pieces to fit conveniently on a page. The connector symbol also provides a way of joining convergent lines of flow that fan into some particular point or of identifying divergent lines of flow.

The terminal connector symbol indicates a beginning, an end, or a break in the usual line of flow. In the first two uses, it is a substitution for an ordinary connector at the beginning and end of major portions of a sequence of symbols (a flow), particularly when these portions are identified by a name, as for a closed program. In its third use, it may represent, for example, a start, a delay, a pause, or an interrupt. For this use, it has both an entry and an exit flowline.

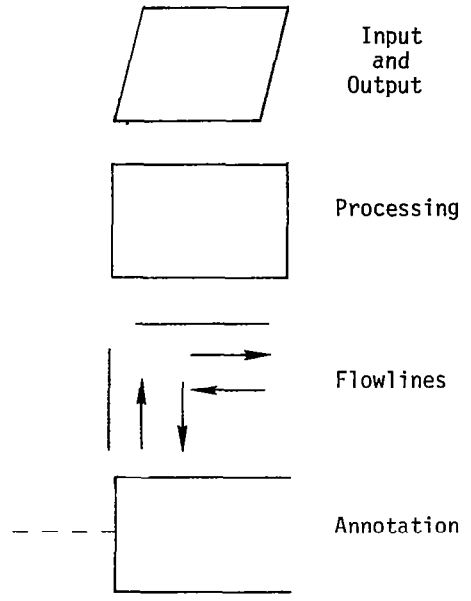
The parallel mode symbol is a pair of horizontal lines with one or more vertical entry flowlines and one or more vertical exit flowlines. It is used to indicate the start or the end of simultaneous operations.

3.2.2.3 Specialized Symbols

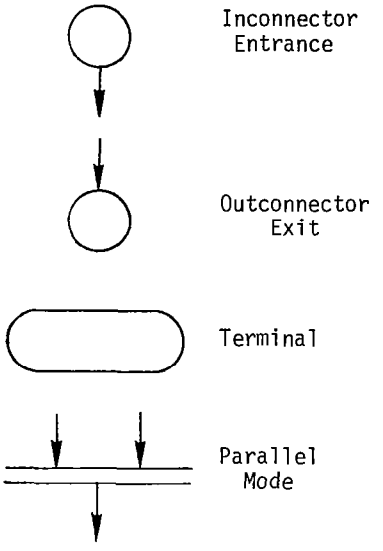
The specialized symbols are not discussed in detail here (see Figure 2), except to point out that the decision symbol (Figure 2(c)) is probably the most widely used and the predefined process symbol (Figure 2(c)) is used for calls to subprograms. For a discussion on the specialized symbols, see Reference 1.

REFERENCE

1. Chapin, Ned: "Flowcharting with the ANSI Standard: A Tutorial," Computing Surveys, June 1970.

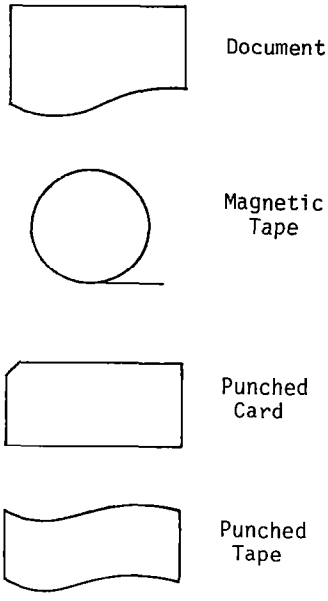


(a) Basic symbols

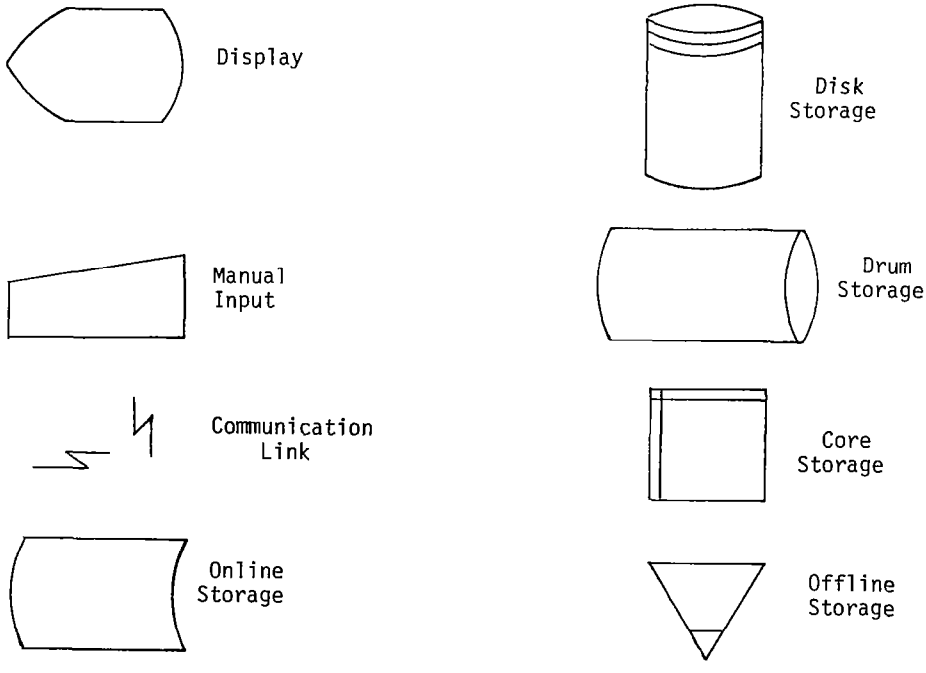


(b) Additional symbols

Figure 1. Flowchart symbols.

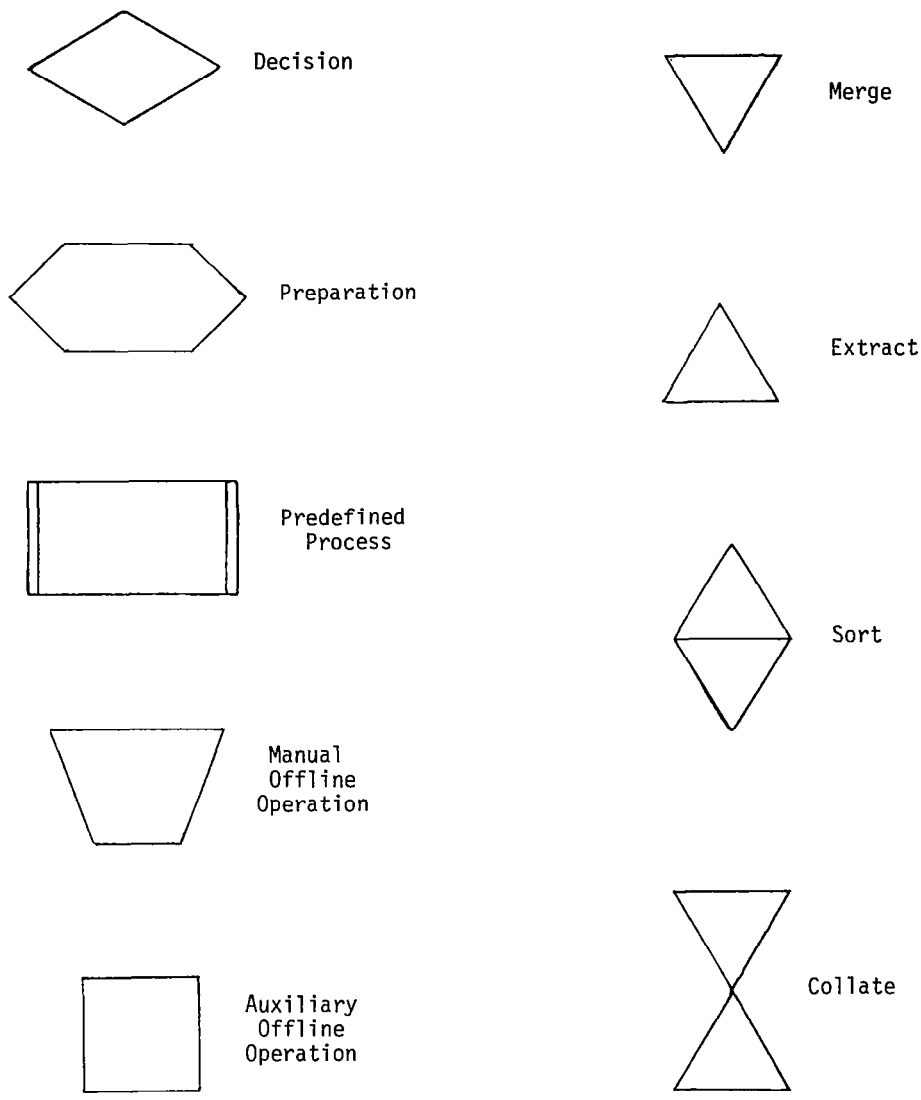


(a) Media symbols



(b) Equipment symbols

Figure 2. Specialized flowchart symbols.



(c) Processing symbols

Figure 2. Specialized flowchart symbols (continued).

3.3 COMMENTS ON STYLE AND EDITING

Entire books are written on style and editing. A classic is Reference 1, so these few words are clearly far from definitive. The principal standard that the writer should try to follow is to meet the needs of the reader. The writer is most effective when he or she is reader-oriented and not self-oriented.

Authors should become familiar with the format specifications for the manuals that are given in Section 3.2 above. Following these specifications should eliminate annoying and distracting inconsistencies and thus help meet the needs of the reader.

The NASTRAN manuals have been designed to accommodate future additions and modifications. Each two-digit section stands alone insofar as it has its own page numbers, equation numbers, figure numbers, and table numbers. Thus, new sections can be added without significant disruption, and entire sections can be added easily (adding sections shall be approved by the NCM in a Special Purpose IRD).

As far as editing is concerned, it is suggested that the NCC appoint an overall documentation coordinator/editor to ensure the new capability is documented consistently across all the deliverables, including the NASTRAN manuals, and is in conformance with the specifications in Section 3.2 above. If organizationally feasible, it is suggested that an editor be appointed for each manual.

REFERENCE

1. Strunk, William Jr. and White, E. B.: The Elements of Style, Second Edition, MacMillan, New York, 1972.

4. ADDING TO AND MODIFYING NASTRAN CODE

4.1 PROGRAMMING IN THE NASTRAN ENVIRONMENT

The New Capability Contractor (NCC) will use the existing NASTRAN framework. Specifically, GINØ, open core techniques, matrix packing routines, and existing NASTRAN utilities shall be used to the greatest extent possible. This list may be expanded in a Special Purpose IRD.

To program in the NASTRAN environment, a detailed knowledge is needed of the use of GINØ, open core techniques, and matrix packing routines. Sections of The NASTRAN Programmer's Manual that significantly affect the programming effort and about which the programmer must become knowledgeable are shown in Table 1.

The NCC shall not modify any existing NASTRAN utility subprogram except

1. TTLPGE (Section 3.3.13 of The NASTRAN Programmer's Manual)
2. INPUT (Section 5.3.2 of The NASTRAN User's Manual)

No other routines in Section 3 of The NASTRAN Programmer's Manual or modules in Section 5 of The NASTRAN User's Manual shall be modified by the NCC. The reason for this policy is that an NCC might modify an existing routine to adequately meet a new capability requirement but at the same time embed an undetected bug that might show up later and render current existing capability inoperative. Of course, this policy does not preclude an NCC from using an existing NASTRAN utility as a baseline, changing its name, making some minor changes, checking it out, and documenting it as a new utility.

Table 1. Key Sections of The NASTRAN Programmer's Manual

1	NASTRAN PROGRAMMING FUNDAMENTALS
2.1	INTRODUCTION
2.2	DATA BLOCK DESCRIPTIONS - GENERAL COMMENTS AND INDEXES
2.4	EXECUTIVE TABLE DESCRIPTIONS
2.5	MISCELLANEOUS TABLE DESCRIPTIONS
3.1	INTRODUCTION
3.2	ALPHABETICAL INDEX OF ENTRY POINTS FOR SUBROUTINE DESCRIPTIONS
3.3	EXECUTIVE SUBROUTINE DESCRIPTIONS
3.4	UTILITY SUBROUTINE DESCRIPTIONS
3.5	MATRIX SUBROUTINE DESCRIPTIONS
4.1	GENERAL COMMENTS AND INDEXES
5	NASTRAN - OPERATING SYSTEM INTERFACES
6	MODIFICATIONS AND ADDITIONS TO NASTRAN

4.2 NASTRAN PROGRAMMING STANDARDS

A subset of the FØRTRAN IV language shall be the language used by the NCC to implement new code in NASTRAN. Assembly language shall not be used except under very limited conditions, which are described in Section 4.2.7.

4.2.1 FØRTRAN Language Restrictions

The specifications governing programming in FØRTRAN IV for NASTRAN are incorporated in the IBM Systems Reference Library manual (Reference 1). Table 1 lists exceptions to the specifications in the manual. It is partially based on the list in Section 6.2 of The NASTRAN Programmer's Manual.

4.2.2 Comments

Frequently, after a programmer is introduced to a system by means of manuals and after a programmer becomes familiar with the system, the manuals are seldom referenced; the source code is the place where programmers go for up-to-date information. Also, the publication of formal, written, external (to the source code) documentation frequently lags considerably behind the source code. Thus, for a long time the ultimate documentation for computer programs has been the source code itself. Until recently, the software industry has been reluctant to acknowledge this. This documentation modus operandi is acknowledged as being natural and effective, i.e., the source code is and should be the primary documentation.

Because comments are so extraordinarily important as an aid to a clear understanding of the source code, the NCC's comments shall be accurate, complete, and easy to understand. Complete sentences using clear, straightforward English shall be used. There should be no misspellings.

Comments are divided into two categories: those in the prologue and those in the code.

4.2.2.1 Comments in Prologue

A prologue is a block of comments at the beginning of each subprogram; each subprogram must have one. Writing a prologue before starting to code helps clarify program requirements for the programmer and makes essential information easily accessible.

A bound set of prologues can make up most of Sections 3.3 and 3.4 of the Design Specification (see Section 2.3 of this document).

Table 2 gives the items that shall be documented in each subprogram's prologue. Items numbered 4 through 8 in parentheses shall be used only for drivers for DMAP modules. A sample use of comments in the prologue is illustrated in Figure 1. Every line begins with a C in column 1. Each heading (ENGLISH NAME, PURPØSE, etc.) begins in column 5; continuations of items begin in column 10. There is a blank line (started with a C in column 1) between each item.

The paragraphs below explain each item in Table 2.

1. ENGLISH NAME - Give the English language name or phrase from which the subprogram name was derived. For example, if the subprogram name is SDR2B, the English language name would be Stress Date Recovery -- Phase 2, Part B.

2. SUBPRØGRAM-MØDULE RELATIØNSHIP - A subprogram can be one of five types: (1) a functional module driver, (2) part of a functional module, (3) a general purpose subprogram, (4) a general purpose subprogram driver, or (5) part of a general purpose subprogram. A functional module driver is a subroutine to which the NASTRAN executive system transfers control if a functional module is requested by the DMAP sequence. Any subprogram that is unique to a functional module is considered as part of a functional module. Any subprogram that performs general purpose functions such as matrix manipulations for more than one module is a general purpose subprogram. Any general purpose function that requires more than one subprogram consists of a general purpose subprogram driver. All other required subprograms are considered part of a general purpose subprogram. General purpose subprograms are documented in Section 3.4 and 3.5 of The NASTRAN Programmer's Manual. Executive subprograms, which are documented in Section 3.3 of The NASTRAN Programmer's Manual, are not considered to be within the domain of the NCC. The existing NASTRAN executive subprograms are considered adequate for the needs of the NCC, and therefore the NCC shall not add to them. The format of item 2 will vary depending on the type of the subprogram:

   1. SUBPRØGRAM-MØDULE RELATIØNSHIP -- THIS SUBPRØGRAM IS A DRIVER FØR MØDULE XXXXXX
      (where XXXXXX is the DMAP name of the module)

   2. SUBPRØGRAM-MØDULE RELATIØNSHIP -- THIS SUBPRØGRAM IS A PART ØF MØDULE XXXXXX
      (where XXXXXX is the DMAP name of the module)

   3. SUBPRØGRAM-MØDULE RELATIØNSHIP -- THIS SUBPRØGRAM IS A GENERAL PURPØSE SUBPRØGRAM

4. SUBPRØGRAM-MØDULE RELATIØNSHIP -- THIS SUBPRØGRAM IS A DRIVER FØR GENERAL PURPØSE SUBPRØGRAM XXXXXX (where XXXXXX is the name of the subprogram)

5. SUBPRØGRAM-MØDULE RELATIØNSHIP -- THIS SUBPRØGRAM IS A PART ØF GENERAL PURPØSE SUBPRØGRAM XXXXXX (where XXXXXX is the name of the subprogram)

3. PURPØSE - Describe briefly, preferably in one or two sentences, what this subprogram does; do not tell how it does it -- this is described under METHØD, No. 9, below.

4. CALLING SEQUENCE - Give the calling sequence to the subprogram. Use the dummy arguments exactly as they are defined in the SUBRØUTINE or FUNCTIØN statement. For DMAP module drivers, give the DMAP calling sequence, following the specifications given in Section 4.1.1 of The NASTRAN Programmer's Manual.

5. INPUT VARIABLES - Describe each input-only variable in the calling sequence. Give the name; description; domain (i.e., lower and upper limits), if applicable; and units, if applicable. For DMAP module drivers, give the input data blocks, following the specifications given in Section 4.1.1 of The NASTRAN Programmer's Manual.

6. ØUTPUT VARIABLES - Describe each output only variable in the calling sequence. Give the name; description; range (i.e., lower and upper limits), if applicable; and units, if applicable. For DMAP module drivers, give the output only (not appended data blocks) data blocks following the specifications given in Section 4.1.1 of The NASTRAN Programmer's Manual.

7. INPUT/ØUTPUT VARIABLES - Describe each calling sequence variable that is both input and output (e.g., a counter that is incremented). Give the name; description; domain and range, if applicable; and units, if applicable. For DMAP module drivers, give the output data blocks that are appended, following the specifications given in Section 4.1.1 of The NASTRAN Programmer's Manual.

8. CØMMØN VARIABLES - For each variable in each labeled CØMMØN, give the block name; variable name; description; an indication whether this variable is input, output, or input/output; domain or range, as applicable; and units, if applicable. For DMAP module drivers, describe the DMAP parameters, following the specifications given in Section 4.1.1 of The NASTRAN Programmer's Manual.

9. METHØD - Describe how the routine fulfills its purpose. Try to be as brief as possible. If applicable, reference previously published material such as texts, papers, NASTRAN manuals, etc.

10. DESIGN REQUIREMENTS - Describe requirements, restrictions, tacit design assumptions, and special features. Points that will be covered include number of scratch files, allocation of core storage, overlay environment, CØMMØN block communications philosophy, input/output assumptions (e.g., which files must be open).

11. DIAGNØSTIC MESSAGES/TESTS - List all user and system message numbers that are in the subprogram. Include a list of all basic tests (such as geometry checks for particular elements).

4.2.2.2 Comments in Code

Comments will be interspersed throughout the code to allow a reader to follow the logic flow and determine the purpose of the various parts of the routine. Sufficient comment statements will be placed at each major switching statement (IF, computed GØ TØ, CALL, etc.) to help the reader understand the flow of the routine.

Variables used for several purposes should have each use noted, or new variable names EQUIVALENCEd to the initial name. When variables, used as constants, invoke a limit (e.g., a maximum value), a comment indicating the restriction should appear in the code, preferably at the time the variable is initialized.

The programmer should generate all the comments as the coding proceeds, while the meanings of the statements are fresh in mind.

It is useful to accompany any debugging statements with appropriate comments to aid in the future debugging of the routine. Such debugging (or diagnostic) statements can be left in the symbolic element and suppressed in the final compilation by use of a C in column 1.

It is suggested that a comment statement, or series of comments, be preceded and followed by a blank comment statement to create the illusion of paragraphing the code.

A final word of caution: common sense dictates that not every line should be commented.

4.2.3 Error-Handling Techniques and Guidelines

Messages will have the following format:

$$*** \begin{Bmatrix} \text{SYSTEM} \\ \text{USER} \end{Bmatrix} \begin{Bmatrix} \text{FATAL} \\ \text{WARNING} \\ \text{INFØRMATIØN} \end{Bmatrix} \text{MESSAGE msgno, text.}$$

where the three asterisks are part of the message; SYSTEM or USER describes the programmer or user, respectively, as being the primary recipient of the message; FATAL, WARNING, or INFØRMATIØN categorizes the severity of the message; MESSAGE is a fixed part of every message's text, which should be as specific as possible and, if a user message, should be as user-oriented as possible; msgno is the message number; text is the text of the message.

Error message numbers (msgnos) will be assigned in blocks by a Special Purpose IRD. For potential Preface error messages, a separate block of error messages will be specified.

Messages that are added to the NASTRAN system by the NCC may be written by the subprogram that detects the error. Use of messages already available in NASTRAN by calls to subroutine MESAGE (see Section 3.4.25 of The NASTRAN Programmer's Manual) should be used if possible.

Although a "fatal" error has taken place, EXIT must not be called; instead, as much processing as possible should be completed. A call to subroutine MESAGE indicating a fatal message should then be made, followed by a RETURN to XSEMi.

4.2.4 Use of the System Input and Output Files

Only Preface modules may read the system input file. This rule implies that any new cards in the NASTRAN data deck must be read by the Preface module designed for that function (e.g., XSØRT will read the Bulk Data Deck).

Routines developed by the NCC may access the system output file only to write error messages (see Section 4.2.3). All other user-oriented data will be formatted by module ØFP.

4.2.5 Single Precision Versus Double Precision

All new modules to be developed and modifications to be made to existing modules should allow both single precision and double precision calculations. Any exception to this rule must be justified by the NCC and approved by the NCM. The 55th word in labeled CØMMØN /SYSTEM/, SYSTEM(55), should

be used to determine what precision is desired by the user for a module. For general matrix manipulation routines, matrix trailers can be used to govern the precision of the calculation to be performed. Providing for single precision and double precision can be done either within a subroutine or, for large computational routines, in separate subroutines (which should be overlaid). Variables stored as double precision variables must not be referenced as single precision variables (via the FØRTRAN EQUIVALENCE statement) because of the different internal word storage format for single precision and double precision words on certain machines (e.g., UNIVAC 1108). The default precision for NASTRAN on the IBM S/360-370 series is double precision, on the UNIVAC 1100 series is double precision, and on the CDC 6000 and CYBER series is single precision.

4.2.6 Use of Assembly Language

Assembly language will be used only in certain very limited cases to promote a substantial benefit in efficiency. Every use of Assembly language must be explicitly approved by the NCM. The NCC will present a written request stating why the NCC believes Assembly language is required. The NCM will either reject the request or will authorize, in writing, the NCC to provide a detailed design for each Assembly language subprogram authorized. Assembly language routines will be designed to run on the IBM S/360-370s, UNIVAC 1100s, and CYBER 170s. Assembly code may be dependent upon the operating system, resulting in further restrictions that will be specified in a Special Purpose IRD. The NCC will be responsible for delivery of checked out Assembly language programs on all three NASTRAN versions.

REFERENCE

1. IBM Systems Reference Library: IBM System/360 and System/370 FØRTRAN IV Language, International Business Machines Corporation, Document No. GC28-6515-8, 1971.

Table 1. List of Exceptions and Extensions to the FØRTRAN IV Specifications.

1. The magnitude of an integer constant or variable may not be greater than $2^{31}-1$.
2. Subscripted variables will contain no more than three subscripts.
3. A CØNTINUE statement requires a FØRTRAN statement number.
4. The PAUSE statement will not be used.
5. The NAMELIST statement will not be used.
6. Implied DØs in DATA statements will not be used.
7. The last statement of a DØ loop will not be a logical IF statement. It is recommended that it be a CØNTINUE statement. It is also recommended that each DØ loop have its own CØNTINUE statement.
8. BLØCK DATA subprograms may contain only type, (e.g., REAL, INTEGER), DIMENSIØN, CØMMØN, DATA and comment statements.
9. All Hollerith data will be defined only in the form 4H....
10. Octal (Ø or B) or hexadecimal (Z) designations in DATA or FØRMAT statements will not be used.
11. Specification statements will precede any executable statement.
12. The order of specification statements will be as follows:

    CØMPLEX

    DØUBLE PRECISIØN

    REAL

    INTEGER

    LØGICAL

    EXTERNAL

    DIMENSIØN

    CØMMØN

    EQUIVALENCE

    DATA
13. Variables in CØMMØN will be ordered as follows: complex, double precision, real, integer, and logical.
14. Variables stored as single precision cannot be referenced as double precision variables (by using the FØRTRAN EQUIVALENCE statement) because of the different internal word storage format for single and double precision words on the UNIVAC 1100 series.

Table 1. List of Exceptions and Extensions to the FØRTRAN IV Specifications (continued).

15. Caution must be exercised to ensure that types (REAL, INTEGER, etc.) of FØRTRAN function values agree in the function subprogram and in the calling subprogram. This agreement between types is necessary for machines (e.g., IBM S/360) on which REAL and INTEGER values of FØRTRAN functions are returned in different registers.
16. No attempt to extend the length of arrays through the EQUIVALENCE statement will be made.
17. Caution must be exercised when using the EQUIVALENCE statement. One should not use the EQUIVALENCE statement to give different variable names to the same word, because modern compilers, due to their optimization techniques, do not guarantee that the values of the EQUIVALENCEd variables will be the same. Hence, EQUIVALENCE should be used only between variables that have nonintersecting use spans in a program.
18. Nonstandard returns in a SUBRØUTINE statement must immediately follow the left parenthesis that starts the names of the subroutine's arguments, e.g., SUBRØUTINE XYZ (*,*,A,B) is the correct form; SUBRØUTINE XYZ (*,A,*,B) is not acceptable. On the CDC computers, use

    SUBRØUTINE XYZ (A,B),RETURNS(RETURN1,RETURN2)

19. There must be agreement with respect to the number of arguments and the type of each argument in the argument list of a calling subprogram and the subprogram called.
20. Deck (or member) names for subprograms will agree with the entry point name. Deck names for BLØCK DATA subprograms will end with the characters BD.
21. FUNCTIØN subprograms whose type is not implicit must be included in the FUNCTIØN statement. For example, use

    DØUBLE PRECISIØN FUNCTIØN ABC (X)

    and not

    FUNCTIØN ABC(X)
    DØUBLE PRECISIØN ABC

22. The name of a FUNCTIØN subprogram must appear somewhere within the subprogram.
23. All subscripted variables appearing in EQUIVALENCE statements must be subscripted, e.g., use EQUIVALENCE (A(1),X(1)) instead of EQUIVALENCE (A,X).
24. DØ loop indexes may not be greater than 2^{17} -1 (131,071). (This is a CDC FØRTRAN (FTN) compiler restriction.)
25. Logical operations are permitted only if using supplied NASTRAN functions, ANDF, ØRF, etc. (See Section 3.4.1 of The NASTRAN Programmer's Manual.)

Table 1. List of Exceptions and Extensions to the FØRTRAN IV Specifications (continued).

26. Subscripts may not contain subscripted variables.
27. Variables in DATA statements will be limited to 4H....
28. DATA statements for variables in CØMMØN blocks must be defined in BLØCK DATA subprograms.
29. Blank CØMMØN will be used only for communication of DMAP parameters.
30. ENCØDE, DECØDE or similar installation- or machine-dependent routines will not be used.
31. Branching into the range of a DØ statement is not allowed.
32. Modification of the length of an explicitly typed variable will not be used, e.g., S/360-dependent type statements such as REAL*8 and INTEGER*2 will not be used.
33. Subscripted variables residing in open core must not be passed by using an argument list. (This is a UNIVAC 1100 restriction.)
34. Multiple entry points in a subroutine are not to be used unless approved by the NCM and the maintenance contractor.

Table 2. Items to be Documented in a Subprogram's Prologue.

1. English Name
2. Subprogram-Module Relationship
3. Purpose
4. Calling Sequence (DMAP Calling Sequence)
5. Input Variables (Input Data Blocks)
6. Output Variables (Output Data Blocks)
7. Input/Output Variables (Input/Output (Appended) Data Blocks)
8. CØMMØN Variables (DMAP Parameters)
9. Method
10. Design Requirements
11. Diagnostic Messages/Tests

```
      SUBRØUTINE GPWG
C
C   ENGLISH NAME
C        GRID PØINT WEIGHT GENERATØR
C
C   SUBPRØGRAM-MØDULE RELATIØNSHIP
C        THIS SUBPRØGRAM IS A DRIVER FØR MØDULE GPWG.
C
C   PURPØSE
C        GPWG CØMPUTES THE CENTER ØF MASS ØF THE STRUCTURE RELATIVE TØ A
C        GIVEN PØINT AND FINDS THE PRINCIPAL MØMENTS ØF INERTIA ABØUT THE
C        CENTER ØF GRAVITY.
C
C   CALLING SEQUENCE
C        GPWG       BGPDT,CSTM,EQEXIN,MGG/ØPGWG/V,Y,GRDPNT/V,Y,WTMASS $
C
C   INPUT VARIABLES
C        BGPDT      BASIC GRID PØINT DEFINITIØN TABLE
C        CSTM       CØØRDINATE SYSTEM TRANSFØRMATIØN MATRICES
C        EQEXIN     EQUIVALENCE BETWEEN EXTERNAL GRID ØR SCALAR NUMBERS
C                   AND INTERNAL NUMBERS
C        MGG        PARTITIØN ØF MASS MATRIX G SET
C
C   ØUTPUT VARIABLES
C        ØGPWG      GRID PØINT WEIGHT GENERATØR ØUTPUT TABLE
C
C   INPUT/ØUTPUT VARIABLES
C        NØNE
C
C   BLANK CØMMØN VARIABLES
C        GRDPNT     INPUT,INTEGER,DEFAULT = -1 - GRDPNT SELECTS THE
C                   GRID PØINT ABØUT WHICH THE INERTIAS WILL BE CALCU-
C                   LATED.  IF GRDPNT IS NØT THE EXTERNAL ID ØF A
C                   GEØMETRIC GRID PØINT, THE BASIC ØRIGIN IS USED.
C
C        WTMASS     INPUT,REAL,DEFAULT = -1.0 - WTMASS GIVES THE
C                   RATIØ ØF MASS TØ WEIGHT FØR THE STRUCTURE.  ALL
C                   ØUTPUT QUANTITIES ARE IN WEIGHT UNITS.
C
C   METHØD
C        THE GRID PØINT WEIGHT GENERATØR MØDULE CALCULATES THE MASSES,
C        CENTERS ØF GRAVITY AND MØMENTS ØF INERTIA ØF THE GENERAL
C        MATHEMATICAL MØDEL ØF THE STRUCTURE.  THE DATA ARE EXTRACTED
C        FRØM THE MATRIX MGG BY USING A RIGID BØDY TRANSFØRMATIØN CAL-
C        CULATIØN.  THE TRANSFØRMATION IS DEFINED BY THE GLØBAL CØØR-
C        DINATE DISPLACEMENTS RESULTING FRØM UNIT TRANSLATIØNS AND
C        RØTATIØNS ØF THE WHØLE BØDY ABØUT A REFERENCE PØINT.
C
C   DESIGN REQUIREMENTS
C        GPWG REQUIRES FØUR SCRATCH FILES.
C
C   DIAGNØSTIC MESSAGES/TESTS
C        THE FØLLØWING FATAL MESSAGES MAY ØCCUR - 3007,3008
```

Figure 1. Sample use of comments in the prologue.

5. DICTIONARY

5.1 DICTIONARY

Table 1 describes the mnemonics, acronyms, phrases, and other commonly used terms in this document. The first column lists the terms in alphabetical order; the second column provides a definition or description.

Table 1. Mnemonics, Acronyms, and Phrases.

DCR	Documentation Change Report
DIAG	DIAGnostic output requested in the Executive Control Deck
DMAP	Direct Matrix Abstraction Program
GINØ	General INput Output
IRD	Interface Requirements Document
NASTRAN	NASA STRuctural ANalysis
NCC	New Capability Contractor
NCM	NASTRAN Contract Monitor
MC	Maintenance Contractor

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3038		

4. Title and Subtitle	5. Report Date
NASTRAN General Purpose Interface Requirements Document	August 1978
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
W. Keith Brown, Frank J. Douglas, John R. McDonough, and David J. Ziobro	CSC/TR-78/6006
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Computer Sciences Corporation System Sciences Division 3217 N. Armistead Avenue Hampton, VA 23666	NAS1-12969
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14 Sponsoring Agency Code

15. Supplementary Notes

NASA Technical Monitor: James L. Rogers

Progress Report

16. Abstract

This NASTRAN (NASA STRuctural ANalysis) General Purpose Interface Requirements Document (IRD) defines standards for deliverables required of New Capability Contractors (NCCs) and relates these deliverables to the software development cycle. It also defines standards to be followed by NCCs for adding to and modifying the code in the NASTRAN software system and for adding to and modifying the four official NASTRAN manuals: The NASTRAN Theoretical Manual, the NASTRAN User's Manual, The NASTRAN Programmer's Manual, and The NASTRAN Demonstration Problem Manual. It is intended that this General Purpose IRD shall be incorporated by reference in all contracts for a new NASTRAN capability.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
NASTRAN New Capabilities in NASTRAN NASTRAN Documentation	Unlimited - Unclassified Subject Category 62

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	79	$6.00

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

NASA-Langley, 1978