# IAC LEVEL "0" PROGRAM DEVELOPMENT

R. G. Vos
Boeing Aerospace Company
Seattle, Washington

Large Space Systems Technology - 1981
Third Annual Technical Review
November 16-19, 1981

## SUMMARY AND CURRENT STATUS

This paper describes activities and software resulting from NASA Contract NAS5-25767, "Integrated Analysis Capability (IAC) for Large Space Systems." This contract is part of the NASA LSST program supporting effort, with direction by the Goddard Space Flight Center (J. P. Young, Technical Monitor).

The Phase I IAC contract effort produced a pilot computer code and a general development plan. That work was reported on at the previous (1980) LSST Technical Review. The ongoing Phase II effort is scheduled to produce an initial operational capability, designated as IAC Level 1, by the end of CY 82.

In the present paper, the first four figures deal with the current IAC status and some planned technical requirements and objectives. The remainder of the paper reports on the development of an intermediate prototype capability, accomplished during FY 81 and designated as IAC Level "0".

The current status of the IAC development activity is summarized in Figure 1. The listed prototype software and documentation have been delivered, and details have been planned for development of the Level 1 operational system. The planned end product IAC is required to support LSST design analysis and performance evaluation, with emphasis on the coupling of required technical disciplines. A recently formalized requirement is for the long-term IAC to effectively provide two distinct features: 1) a specific set of analysis modules (thermal, structural, controls, antenna radiation performance and instrument optical performance) that will function together with the IAC supporting software in an integrated and user friendly manner and 2) a general framework whereby new analysis modules can readily be incorporated into IAC or be allowed to communicate with it.

- **Ongoing Phase II Contract**
- **Prototype software delivered**
  - Technical modules – MSC NASTRAN®, DISCOS, TRASYS, SINDA, ORACLS
  - Solution paths – standalone, thermal/structural, structural/control, thermal/structural/control
  - Executive/data management/graphics/module interfaces
- **Draft documentation delivered**
  - User manual for prototype system
  - Functional specs document
- **Details planned for operational system**

Figure 1

## IAC REQUIRED CAPABILITIES

Much of the required technical capability of IAC can be described as being part of one or more distinct "solution paths." Each path is actually a class of solutions, which consists of a number of selectable options and variations, rather than a rigidly predefined and automated process. An engineer-in-the-loop mode of operation is therefore possible and, in fact, emphasized. Currently, five such solution paths, as shown in Figure 2, have been defined. The solid lines of paths I to IV indicate capabilities which have been implemented and are available for use and evaluation within the current prototype software package. The standalone (uncoupled) operation of each technology or major technical module is defined to be Solution Path I. Paths II through V involve an increasing degree of interdisciplinary coupling and corresponding greater complexity. Solution Path II provides thermal deformations via the coupling of a thermal analyzer such as SINDA or NASTRAN with a structural analyzer such as NASTRAN or SPAR. A prototype modeling integration module (MIMIC) has been implemented during FY 81 to handle data flow between the generally incompatible thermal and structural models. Path III accomplishes a structural/control analysis, in either the frequency or time domain, by providing required modal data from a structural analyzer to the DISCOS system dynamics module. Solution Path IV has been implemented during FY 81. It provides a time domain thermal/structural/control analysis, including a time varying but quasi-static thermal loading, i.e., thermal loads are unaffected by the dynamic motions. Finally, Path V is to provide a fully coupled analysis in the frequency domain and is directed at problems such as thermal flutter of long spacecraft members.

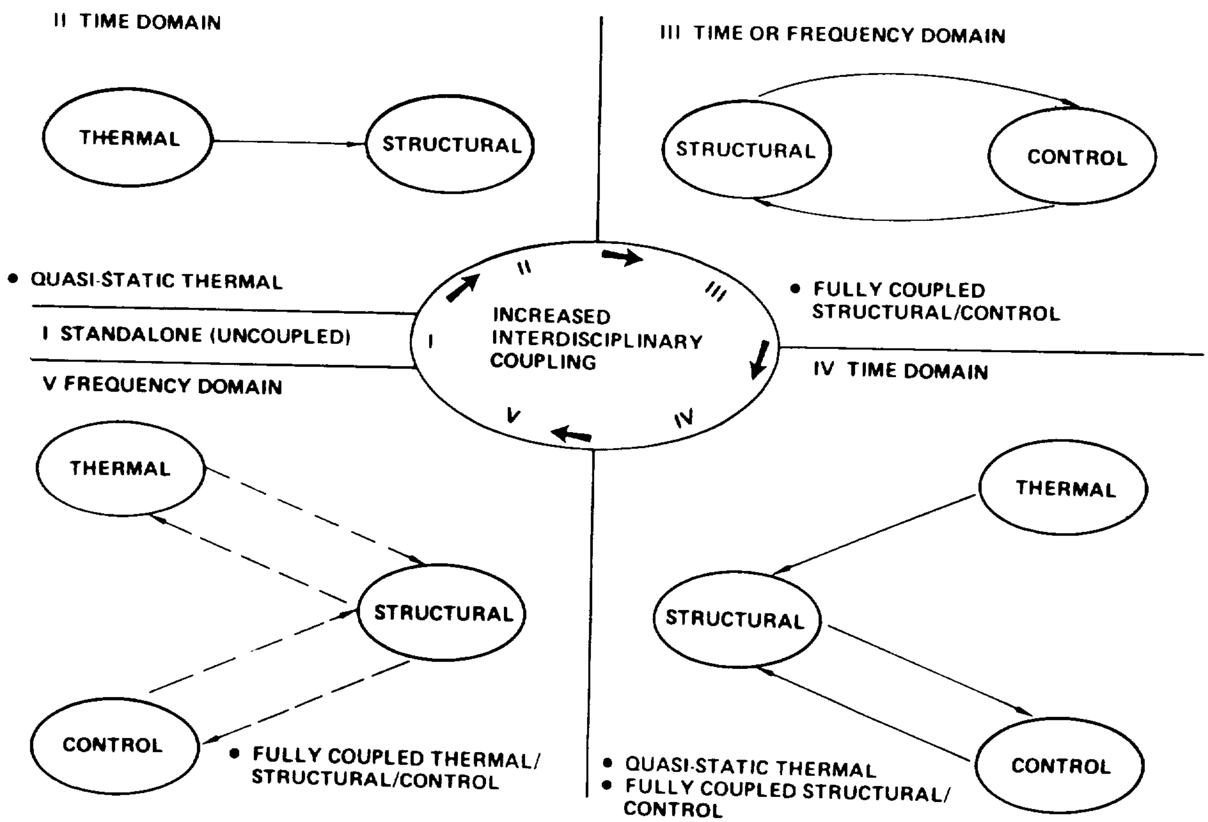



Figure 2

## IAC SELECTED TECHNICAL MODULES

The technical modules currently defined for implementation within IAC are shown in Figure 3. These modules are classified into four technical groups - system dynamics, structural, thermal and controls. The solid lines indicate capabilities which have been incorporated into the prototype software system, while dashed lines indicate planned capabilities which have not yet been implemented. DISCOS (Dynamic Interaction Simulation of COntrols and Structures) is a primary computational backbone of the IAC system. The selected thermal and structural modules are generally well known within the technical community. ORACLS (Optimal Regulator Algorithms for the Control of Linear Systems) is a package of selected subroutines which emphasizes modern control theory design. SAMSAN (SAMpled System ANalysis capability) is a similar package which emphasizes classical control methods and is currently under development at GSFC. The MODEL controls program will consist of several special-purpose interactive or batch versions, which in general create FORTRAN code to numerically solve a set of user defined differential equations.

It will be readily apparent to those familiar with the designated structural and thermal modules that there is some duplication of capability, e.g., NASTRAN/SPAR and SINDA/NASTRAN. This is due in part to a Phase I study and conclusion that both finite difference and finite element thermal codes should be available within IAC. More importantly, it is the result of a conscious effort to provide alternate technical modules within several areas of IAC in order to support as wide an existing user community as practical. The list of IAC technical modules will continue to grow as additional user groups define LSST requirements and as technology and data-coupling are implemented in areas such as antenna radiation and instrument optical performance.

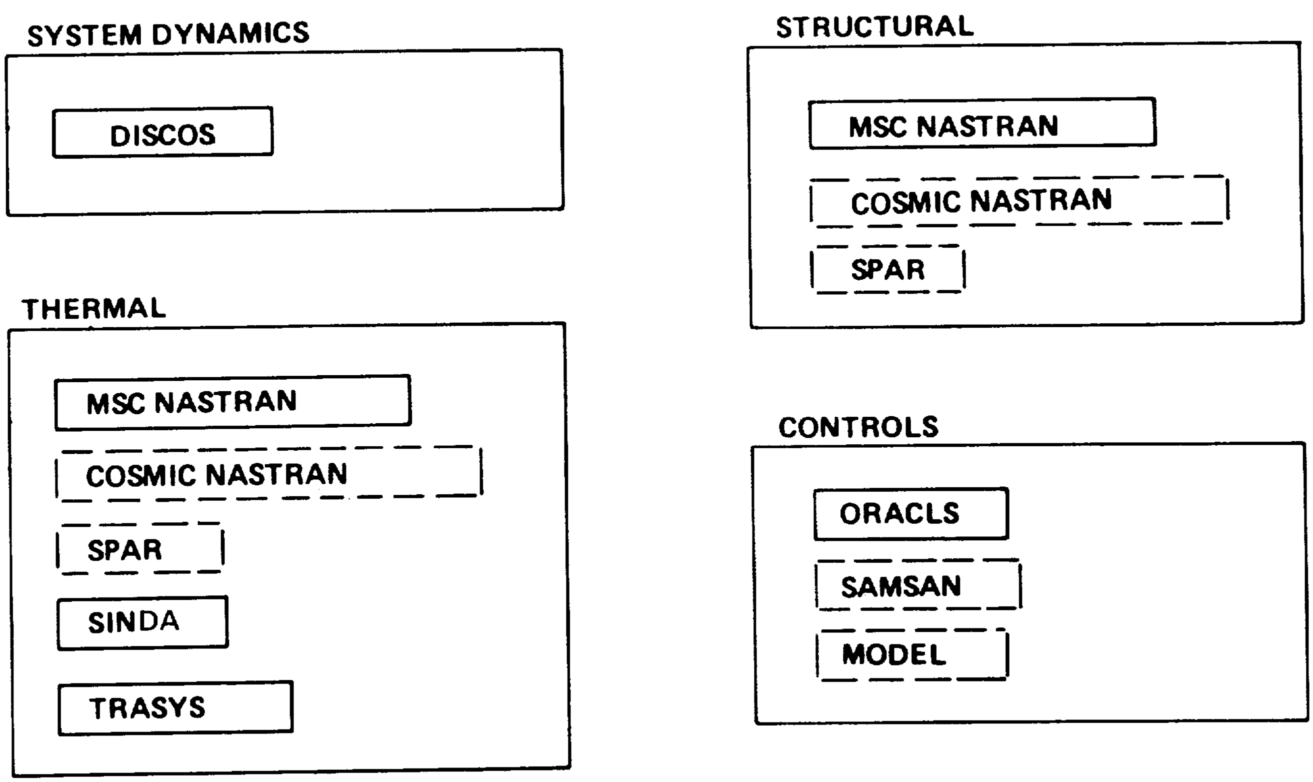



Figure 3

## IAC SYSTEM INTEGRATION PHILOSOPHY

The solution paths already discussed define basic technical requirements of the IAC system. The selected technical modules provide major components for supporting the individual technologies represented within these paths, and required interdisciplinary couplings are largely being implemented via new interface software which bridges between the different technologies and mathematical modeling techniques.

At the overall system level, Figure 4 summarizes the philosophy for integration of the entire IAC package. Key characteristics required of IAC are shown on the right, and corresponding components of the supporting software and hardware are given on the left. First, the computational complexities inherent in most LSS problems led to an early decision that IAC must operate in an engineer-in-the-loop fashion, rather than in a highly automated "button pushing" mode of operation. A specialized executive program is used to make all capabilities available to the engineer, in a modular but consistent and engineer friendly manner. Second, in order to accomplish in-depth analyses with the selected technical modules, in a reasonably efficient and natural manner, a file oriented data management system has been developed. In order to provide effective user access to data, some enhancements to the file oriented system, relative to data identification and display, have been implemented. For the same reason, considerable emphasis is being given to interactive graphics. The IAC target host computers are state-of-the-art machines with significant virtual memory capability. At the present time such machines are largely in the super-minicomputer class, but new mainframes can be expected to increasingly satisfy this target requirement. The DEC VAX 11/780 super-minicomputer is being used for the current IAC development.

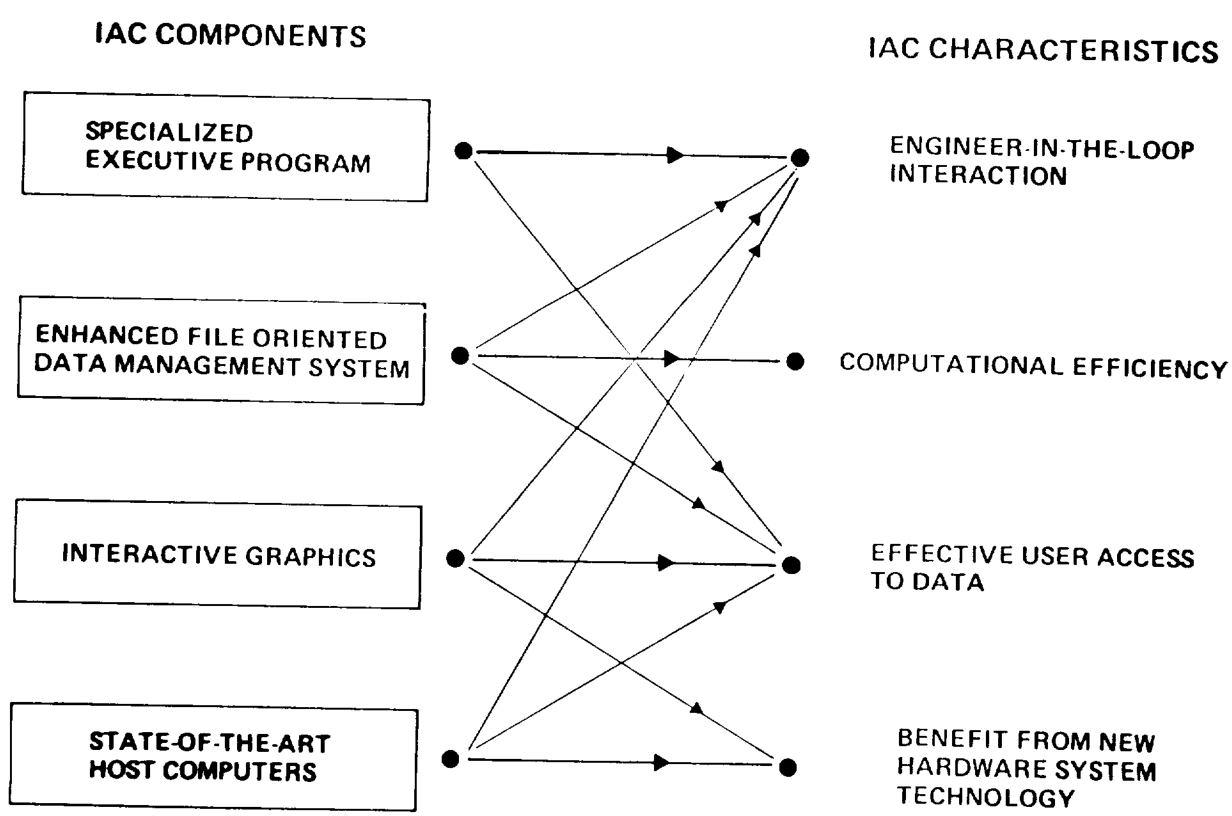



Figure 4

## COMPLETED LEVEL 0 ARCHITECTURE

A schematic of the Level 0 (prototype) architecture is shown in Figure 5. The executive provides the user with a common interface to all of the diverse technical modules and to the IAC supporting software. The executive software includes a command interpreter, module driver, and data handling and graphics display capabilities. The software is programmed in FORTRAN '77, and the graphic capabilities are based on the DI-3000 SIGGRAPH Core standard support package. The executive provides for access to, and communication between, three types of data storage areas: (1) a file oriented database; (2) a user-specific virtual memory workspace; and (3) the host file system. The major technical modules currently implemented within IAC are DISCOS, MSC NASTRAN, ORACLS, SINDA, and TRASYS. The interface modules INDA, etc. provide required data-flow linkages between IAC and the technical modules. The MIMIC interface module is a prototype mesh variable transformation capability designed to aid the user in handling modeling incompatibilities, e.g., between thermal and structural analyses in IAC solution Paths II and IV.

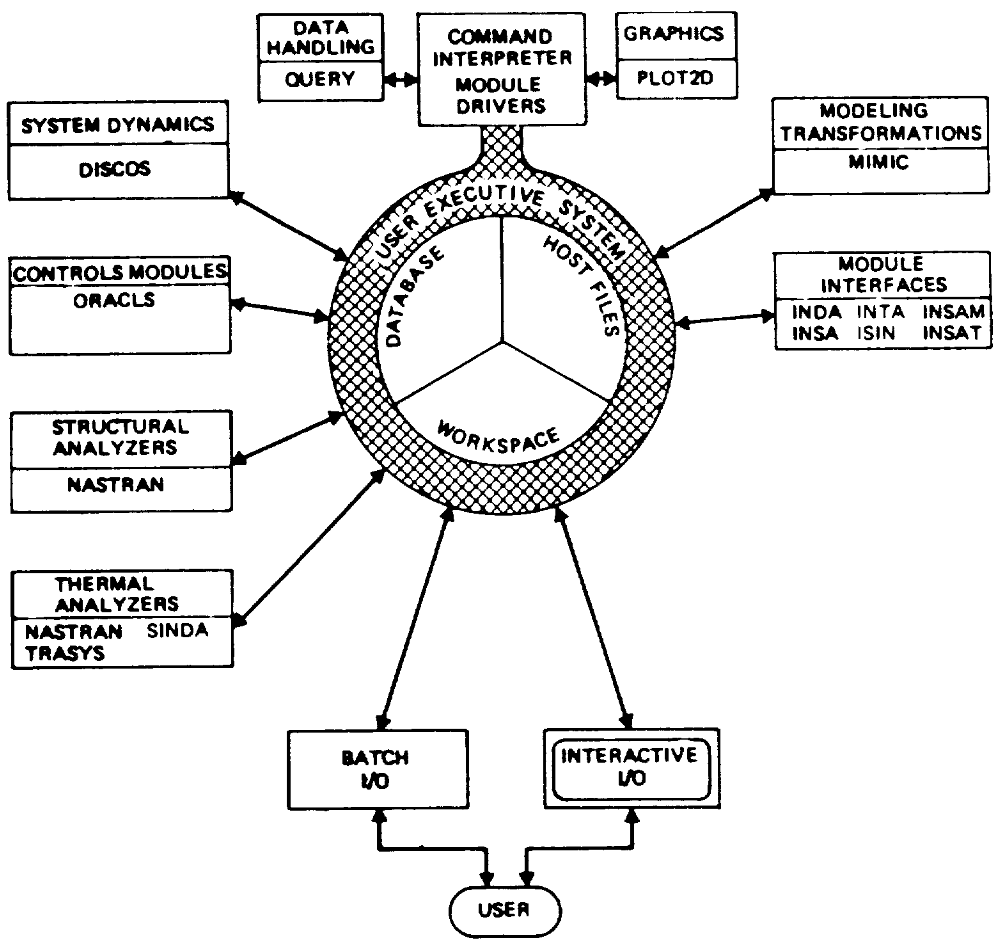
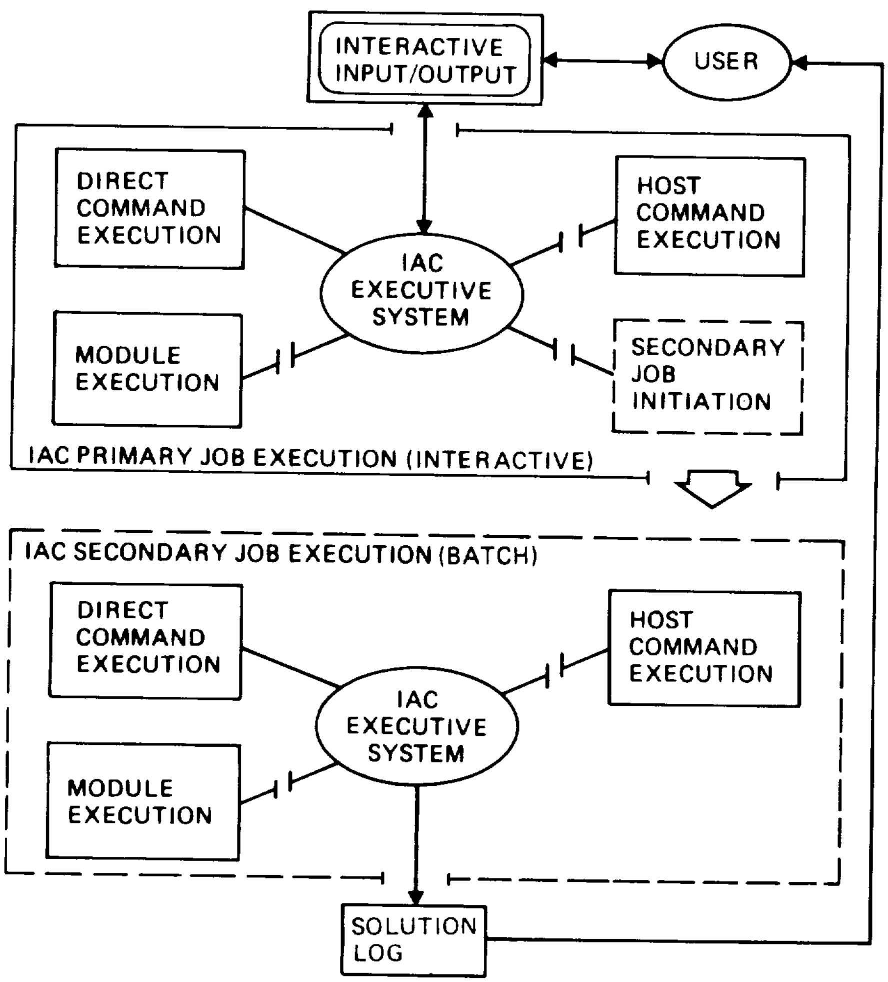



Figure 5

## EXECUTIVE JOB FLOW

The executive integrates other components of the system into a unified package, and provides the primary interface between IAC and the user. A schematic of executive functions is shown in Figure 6. The emphasis is on interactive operation; however, modules are often executed in essentially a batch mode. The user accomplishes the mainstream of his tasks within the IAC "primary job." Within this job he/she may request the executive to execute a module, with user specified parameters. He/she may also request that a sequence of commands (including module executions) be initiated as a separate "secondary job," in a batch mode concurrently with the primary job. In addition, the user may execute many direct commands, e.g., relating to help information, data handling or graphics tasks. Certain direct commands cause the execution of lower level executive routines, which may then be driven by user tutorials or menus. In order to fully utilize the host operating system features, a capability has also been developed to execute any host (computer operating system) command, or sequence of host commands, from within the IAC executive. The broken connecting links in the figure denote temporary transfers of control between IAC and the host operating system.




- EXECUTIVE TASKS
  - INTERPRET COMMANDS
  - EXECUTE DIRECT COMMANDS
  - EXECUTE HOST COMMANDS
  - EXECUTE MODULES
  - INITIATE JOBS
  - PASS PARAMETERS
  - TRANSFER DATA
  - QUERY DATA
  - PLOT DATA

Figure 6

## DATA HANDLING

IAC provides a variety of data handling tools which support data computation, manipulation and evaluation. Figure 7 lists some typical processes which are supported by these tools.

The IAC data handling system has been designed with a primary emphasis on computational efficiency (in generation and use of new bulk analysis results) and a secondary emphasis on informational type query (for processing and display of existing items of information). The host file system is utilized to a large degree to provide an efficient and natural interface to many of the IAC computational modules. The database and associated data structures facilitate communication by providing standard data organizations and formats. The virtual memory workspace provided by the IAC executive allows for detailed user query of data.

IAC currently provides three types of data structures - array, relation, and TOC (table of contents). These data structures may contain integer, real, double precision, and/or text type data. An IAC array consists of a general matrix (arbitrary order and index dimensions), and optional labeling data associated with each index. A typical array of node/time/temperature data is illustrated in Figure 7. Such arrays can be used to communicate between different modules (e.g., provide thermal analyzer results as loadings to a static deformation analyzer). They also permit user query, e.g., display of temperature ranges for a given set of nodes, or plotting of temperature versus position for selected times and nodes. A relation is a 2-dimensional table, where each column has an associated name and each row represents a particular occurrence of values. A TOC is a special form of relation, which catalogs the characteristics of other data structures.

- TYPICAL DATA HANDLING PROCESSES

| | | |
|---|---|---|
| DEFINING | SELECTION/TALLY | MERGING/JOINING |
| LOADING | PRINTING | EXTRACTION/PARTITIONING |
| CHANGING | SORTING | DATA-STRUCTURE TRANSFORMATIONS |
| DELETING | STATISTICAL COMPUTATIONS | HOST FILE INTERFACING |

- TYPICAL ARRAY DATA STRUCTURE

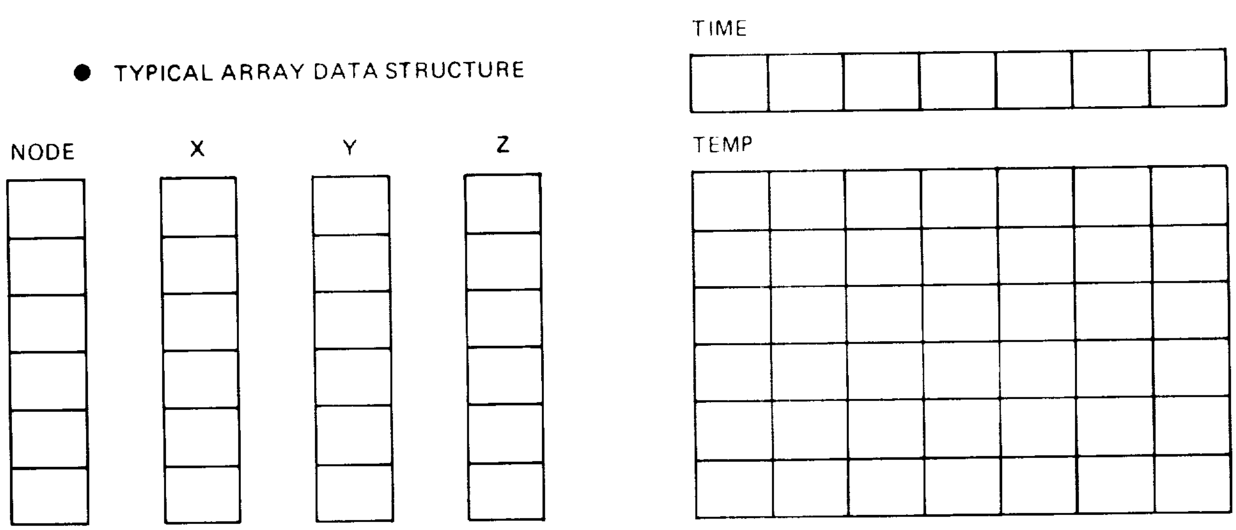



Figure 7

## GRAPHICS DATA DISPLAY

It is useful to classify graphics applications within a system such as IAC into the areas of geometry, math models, graphs, and field descriptions. The Level 0 development has produced a general graphing capability, by which relationships between variables in an IAC database or workspace may be displayed. Geometry and math-model graphics applications can be handled by a variety of available but usually installation dependent packages for CAD and model generation/display (e.g., NASCAD, AD-2000, SUPERTAB, PATRAN). IAC will be capable of interfacing with such packages. IAC will also utilize, wherever possible, the available graphics capabilities of individual technical modules. Field description refers to the display of analytical field variables (e.g., temperature, stress) as a function of time, geometrical or math-model associated position, etc. Recent advancements in color raster graphics hardware are especially applicable to field description type displays.

Figure 8 shows a schematic of the current IAC graphing capability for display of X-Y curves, bar graphs, correlation tables, etc. Data is selected from the database or workspace by the user via a tutorial prompting routine, and a card image plot file is created. The plot file is a complete and device-independent numerical representation of a graph. The plot file provides a useful interface between the raw data and the pictorial display; the plot file can also be directly created or modified by the host editor, and stored or retrieved via the user's host file directory. The actual CRT graph display is generated from the plot file, via a DI-3000 based interactive plotting package. DI-3000 is an ACM SIGGRAPH Core standard graphics support package, which IAC has utilized in its graphics software development.

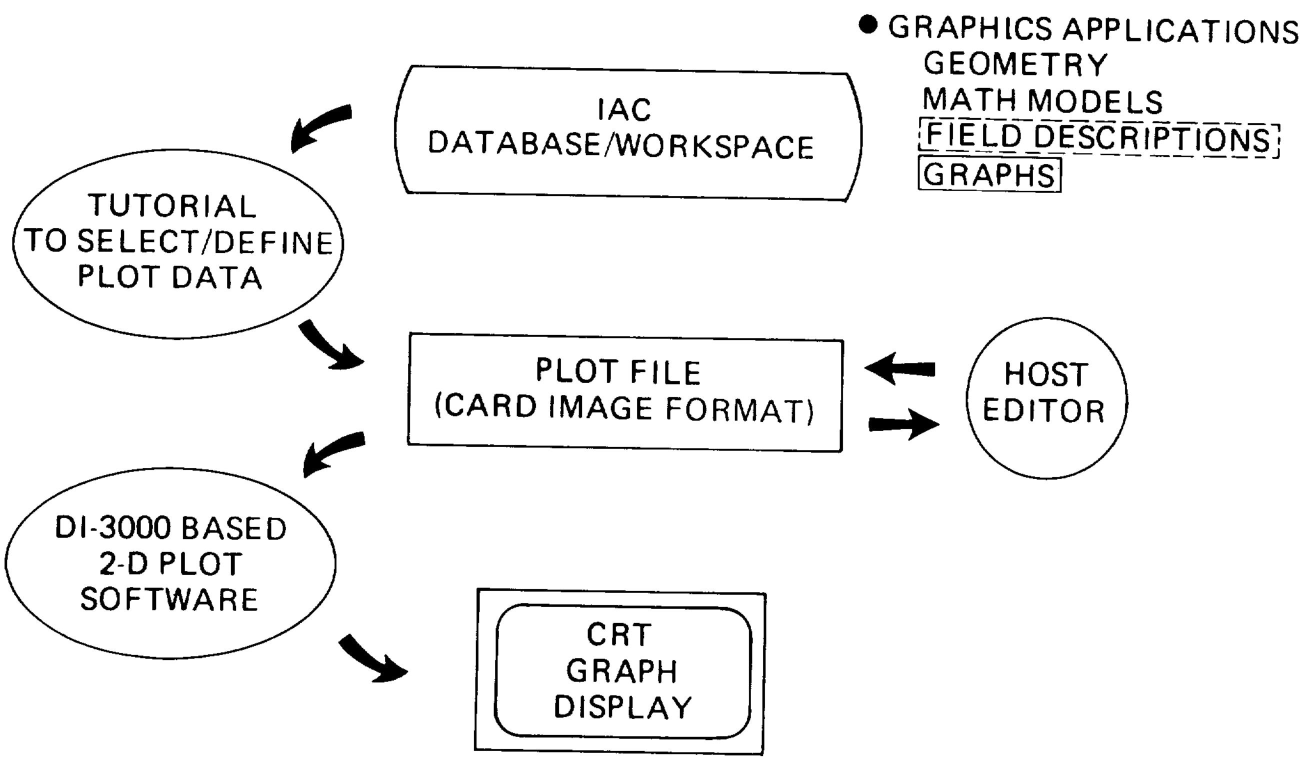



Figure 8

## TECHNICAL MODULE ACCOMPLISHMENTS

The FY 81 accomplishments related to technical modules are summarized in Figure 9. The new MSC NASTRAN Version 61 was implemented within IAC. This required modification of several IAC/NASTRAN interface modules because of some NASTRAN upward incompatibilities in solution procedures and DMAP modules. The DISCOS program was enhanced to include effects of time varying quasi-static thermal and mechanical loadings (IAC thermal/structural/control solution path). The TRASYS radiation analyzer was implemented, with some streamlining in the user/JCL interface and in the handling of TRASYS scratch files. The associated graphics module TRASPLOT was also implemented within IAC. The SINDA module (including the SINFLO fluid flow capability) was implemented with some modifications similar to those for TRASYS. The ORACLS software package was converted from the CDC to VAX host computer, consistent with FORTRAN '77 language syntax. The IAC/ORACLS implementation provides for execution of both standard and user-defined driver programs for selected groups of ORACLS routines. A particular standard driver has been developed for design of a continuous or discrete regulator.

As stated earlier, one of the long-term IAC objectives is to provide a general framework which simplifies the incorporation of new modules into the system. This objective can best be met by an evolutionary process, and some initial results have been achieved during the Level 0 development. A general table-driven set of module execution software has been developed, which reduces the executive code required to implement a new module. Some improved techniques have been devised for handling module scratch storage and output listing file requirements. Standard JCL procedures have been developed, which have general applicability to many FORTRAN '77 oriented modules.

- MSC NASTRAN
  - New version 61 implemented
- DISCOS
  - Mods to include quasi-static thermal loadings
- TRASYS
  - Computational module TRASYS implemented
  - Graphics module TRASPLOT implemented
- SINDA
  - Computational module SINDA implemented
- ORACLS
  - Conversion from CDC to VAX
  - Provision for both standard and user-defined drivers
  - Driver developed for continuous or discrete regulator
- Techniques to simplify incorporation of new modules

Figure 9

## DATA-FLOW ACCOMPLISHMENTS

A primary task of IAC is to integrate (or interface) various existing technologies and capabilities, via establishment of appropriate data flow between them. The FY 81 accomplishments related to data flow are summarized in Figure 10.

Interface modules have been developed or extended for NASTRAN thermal, dynamics and statics solutions. These modules accomplish the transfer of data between NASTRAN and the IAC database, and in some cases provide automated generation for portions of a NASTRAN input file. DISCOS is capable of obtaining input data arrays directly from the IAC database. The Level 0 development has provided the capability for DISCOS to obtain time varying modal thermal displacement data from the database. The IAC/ORACLS implementation emphasizes direct communication with the database and utilizes many of the IAC data handling tools for matrix definition, manipulation and display. The Level 0 development has provided the capability for ORACLS to transfer arbitrary matrices to and from the IAC database. A SINDA interface module has been developed to transfer particular output results to the database. A subroutine has been added to TRASYS which allows radiation results to be created in NASTRAN consistent input format. A mesh-point interpolator (MIMIC) has been implemented which aids in transforming field variables from one math model to another. The interfaces necessary to support the IAC thermal/structural/control solution path have been completed.

Looking to the future, there will undoubtedly be some useful technical modules which do not execute within IAC or communicate with an IAC database. Therefore, some prototype IAC tools have been developed to provide general data flow between module formatted host files and an IAC database.

- IAC module interfaces
  - NASTRAN THERMAL
  - NASTRAN DYNAMICS (normal modes)
  - NASTRAN STATICS
  - DISCOS
  - ORACLS
  - SINDA
  - TRASYS/NASTRAN
- Non-IAC module interfaces
  - Host-file/IAC-database transformation capabilities
- Modeling integration
  - MIMIC (mesh point interpolator) implemented
- Solution Path IV (thermal/structural/control) completed

Figure 10

## THERMAL/STRUCTURAL/CONTROL SOLUTION PATH DEVELOPMENT

The IAC thermal/structural/control solution path was accomplished during FY 81. It provides a time-domain system dynamics analysis, including the effects of time-varying but quasi-static thermal loads (dynamic changes in configuration are assumed not to affect the thermal loading). Four major technical modules are involved in this solution path-- transient thermal, dynamic normal modes, static deformation, and system dynamics (including controls effects).

Operation of the last part of this solution path is illustrated in Figure 11, including only NASTRAN and DISCOS as the last two above mentioned technical modules. (Required data from the transient thermal and the normal modes analyses are assumed to be already existing in the database.) As shown, the user supplies a NASTRAN statics partial input file, containing a nodal-based math model. Interface software is then executed which creates an enhanced input file, using mode shapes and transient nodal temperatures from the database. This process automatically generates NASTRAN thermal load sets, and converts the model from a nodal to a modal basis via an approach similar to static condensation. The NASTRAN statics analyzer is then executed and the computed time-varying modal thermal displacements are stored in the database. Nodal displacements can also be made available for user evaluation and display. A DISCOS time-domain analysis is finally performed, using the modal thermal displacements as quasi-static loadings along with mode shapes and modal characteristics also in the database.

It should be noted that although this solution path is primarily oriented toward quasi-static thermal loadings, it is equally applicable to quasi-static mechanical loadings since general time varying modal displacement loadings are passed to the system dynamics analyzer.

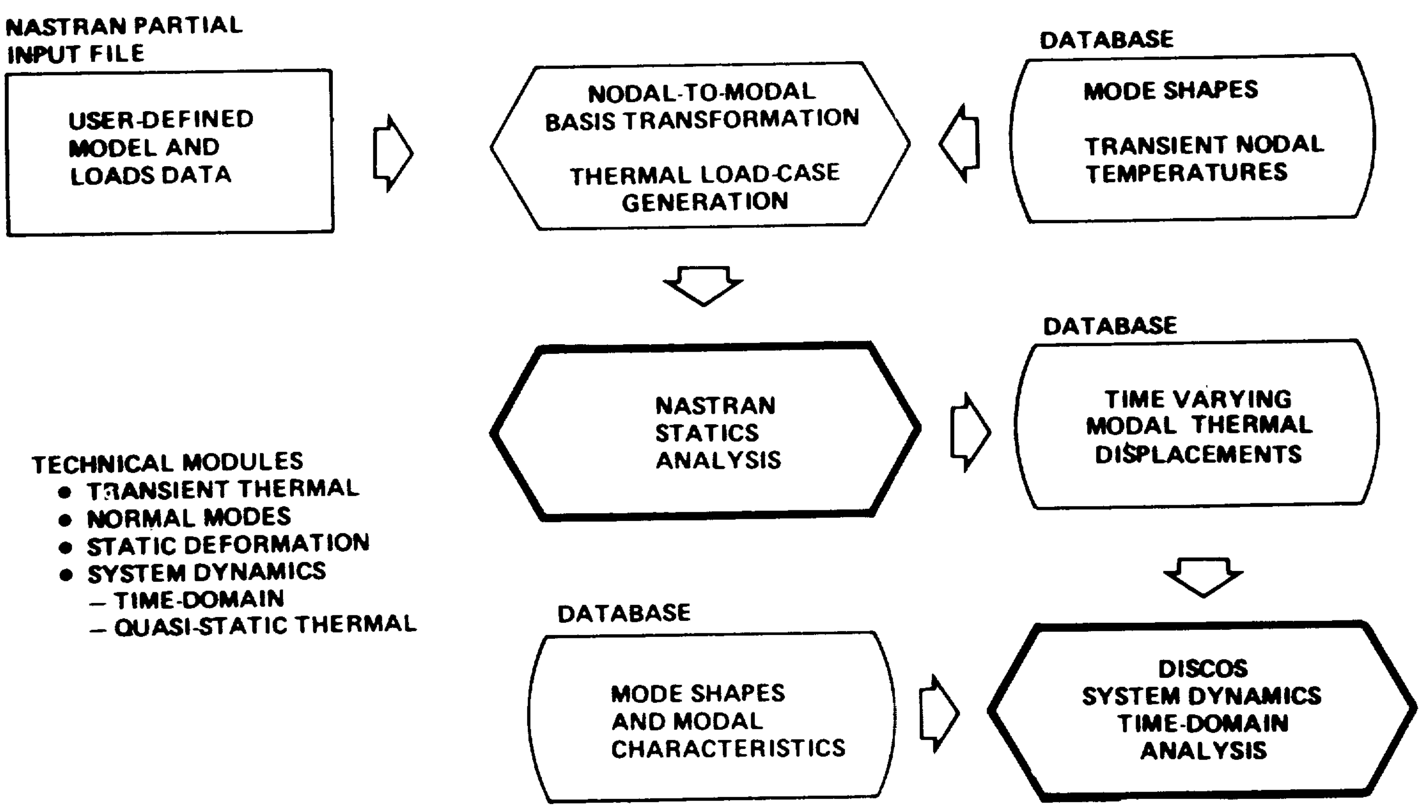



Figure 11