Integrated Structural Analysis for Rapid Design Support

D.G. Wong* and C.R. Fuller†
Lockheed Missiles and Space Company, Sunnyvale, California

This paper describes an integrated structural analysis system that has been developed to enable analysts to have significant influence during the conceptual stages of design and to have confidence in detailed analyses. The system requirements, hardware, and system and application software attributes are discussed in detail, and evidence demonstrating the effectiveness of the analysis system is presented. Information is provided to give the reader some insight into the rationale for hardware acquisition and software development and to develop system design principles applicable to his own situation. A discussion of hardware attributes includes a specific list of equipment, network communication links, and specialized terminals for pre- and postprocessing capabilities. A discussion of software attributes focuses on system architecture, program modularity, coding standards, and data structures for application software. The integrated analysis system described herein has been operating successfully for two years, and design organizations, management, and customers have all recognized its benefits.

Introduction

USING conventional finite-element analysis techniques, the total time required to perform reasonable evaluations of structural designs during preliminary concept studies is such that only a few design concepts and iterations can be evaluated before a final design is selected. This results from the increased design complexity necessary for meeting weight savings and performance goals. Increasing design complexity, which includes the development of composite materials such as graphite/epoxy laminates, precludes the use of simplified analysis methods. These complex designs require state-of-the-art analysis techniques such as finite-element methods to perform dynamic load and stress analyses. For launch vehicle structures, several specialized computer programs are used to produce the governing design load, and many detailed stress models are generated for strength analyses. Frequently, much of the output data from one analysis provides the input data for a subsequent analysis, and many man-hours are expended in the preparation of computer input both in the construction of the mathematical models and in the transfer of data from one analysis to another. These tasks occupy much of the analyst's time, which could be spent more creatively. Frequently the analysis lags the design, which limits design optimization opportunities. The expected complexity of future designs coupled with the importance of design optimization mandates significant reductions in the times for performing evaluations of particular structural concepts.

A system incorporating the latest developments in both computer hardware and software was conceived in accordance with a five-year plan that was initiated to support future advanced launch vehicle programs. The system, completed ahead of schedule, has been operational for two years and has proven to be effective.

Received March 6, 1986; presented as Paper 86-1010 at the AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics and Materials Conference, San Antonio, TX, May 19–21, 1986; revision received Sept. 11, 1986.

*Manager, Structures Department, Missiles and Space Division.
†Group Engineer, Structures Department, Stress and Loads Group.

Engineering Requirements

The initial selection of areas having a significant payoff is described in Ref. 1 and starts with the organized approach of focusing attention on time-consuming operations. As the implementation of the the plan progressed, it became evident that user attitudes were improving and enthusiasm for the system was growing with employee awareness. In retrospect, this became a vital point in that user involvement was found to be essential in defining features having a significant impact on increasing productivity. Eventually, the users became thoroughly convinced of the merits of the program and participated enthusiastically in establishing requirements, as depicted in Fig. 1.

A centralized computer facility that provides engineering services for both batch and interactive users was used. As one may envision, such a facility has many limitations—some of which we believed could be satisfied with a distributed minicomputer system coupled with terminals located at user workstations. Also of paramount importance are the refresh terminal for model debugging and displaying dynamic events and the color terminals for the postprocessing of computer results. The numerical computations required for most FEM models will still be accomplished by the central facility that is located in another building but linked by private cable to the minicomputer. Printers are available for the local postprocessing of results and reports issued with standard or text standard software.

There is a need to provide rapid analyses of test results from facilities at our company's primary location and at remote locations in other cities. As such, a telecommunication link was established along with standardized data formats to facilitate the transfer of data from on-site test facilities to our local computer. Experience shows that the most expedient transfer of data from some test facilities is still with magnetic tape using standardized data formats. Lines were also established with a customer PC across the country and with other PCs within the company using the minicomputer as a central switching and storage hub, such that the PCs would perform as satellite terminals. Every effort was made to standardize the hardware among the company's three engineering divisions, facilitating electronic mail communication and the sharing of system and applications software.

It was decided that software development standardization was necessary for a successful system and that experts from each of the three major divisions should be involved in the standards selection. After considerable thought and numerous meetings, several informal agreements were reached. The central theme of software development focused on the promotion of stand-alone components and the discouragement of complicated, large, monolithic codes. By enhancing the modularity and documentation of software components and by following well-defined ground rules in the design of the global database logical data structures, an environment has been created which is conducive to the interchange of technology among related disciplines. It is recognized that the advent of new computer hardware is constantly reducing the dollar cost of numerical computations while engineering costs are rising. Therefore, meeting user and programmer needs is more significant than maximizing computational efficiency in promoting increased productivity. These needs are being met by the establishment of software development standards that include a global database system that frees the user from the details of physical data structure, a user-friendly command language that permits the integration of different processors, and software coding methods that mimimize and isolate machine dependencies.

With respect to application software, the computation of loads requires several unique nonlinear codes for describing the dynamic responses resulting from depth charge, launcher dynamics, and mechanical separation during staging. In general, these dynamic response codes utilize the physical properties of the missile as represented by modal characteristics. With all the different configurations that must be considered, especially for dynamic controls and stability studies, there is a pressing need for efficient eigenfunction routines, especially for large FEM models. Attendant with the eigenvalue extraction routines, a compatible linear modal acceleration dynamic response package and shear moment eigenvector routine is now operational for the automatic transfer of data among codes. There are also needs for a spectrum response and graphical displays for the comparison of analyses to experimental data, and these are now operational. For stress analyses, it was evident that the final hardware product would benefit directly if the analyses could provide conceptual design support in addition to the usual evaluation of the detailed design. Consequently, the analysis time cycle had to be reduced from weeks to days or at most a week. Model generation and checking, numerical computations, and postprocessing were all time-consuming operations that were addressed to our satisfaction for meeting our commitments. Most models are large, three-dimensional, and linear, but occasionally nonlinear and composite lamination models are required.

CADAM[2] system, links are now available for acquiring the geometric and nodal data for the generation of FEM models. There is also a need to utilize CADAM for the generation of mass properties data such as weights, the center of gravity, and inertias for weight trades and detail computations.

Hardware

The local computer facility consists of two DEC VAXes in a "VAX cluster" configuration. One VAX is an 11-785 with 12 megabytes of main memory, and the other is an 11-780 with 10 megabytes. The VAX cluster arrangement allows the two processors to share a common set of peripherals, including printers, tapes, and disks. The VAX system supports 54 terminals split equally between DEC VT240s and VT100s, along with two AED 512 color displays and one Tektronix 4114 high-resolution storage tube monitor. The VT100s have been modified to provide low-resolution graphics with a Selanar 100 PC card, so virtually all of the terminals on the system have some graphic capability. Also connected to the VAX system are one DEC Rainbow (PC100), three DECMATE IIs (PC278), two DEC Professionals (DEC PC325, DEC PC350), and one IBM PC5160. The DEC PC350 is being used as a standalone computer and has been certified for classified work in an open but secured area within the user community. Two of the three DECMATE IIs (which do not have graphic capability) will soon be phased out and replaced with DEC VT240s to provide graphic capability. In addition, the DEC 325 is being used only as a terminal.

The file system, which is common to the two processors, consists of five DEC RA81 fixed disk drives, two DEC RP07 fixed disk drives, and two DEC RM05 removable pack drives. The RA81 drives are served by an HSC-50 intelligent storage controller. Total on-line storage capacity is over 3.8 gigabytes. One DEC TU77 (800/1600 BPI) tape drive and one DEC TU78 (1600/6250 BPI) tape drive provide off-line storage capability. An RX02 floppy diskette drive is available for intersystem file transfer.

Hardcopy output is provided by two Versatec V-80 electrostatic printer/plotters, one DEC LN01 laser printer, and three letter-quality printers (two DEC LQP02s and one NEC Spinwriter). All of the printers are available from both VAXes. An HP 7220 eight-color pen plotter and a Matrix Model 4007 photographic recorder system provide color hardcopy. A Megatek 7220 graphics subsystem running WAND/LIFE software is attached to one of the VAXes. It provides three-dimensional, real-time, refreshed graphics for simulation and CAE applications. In 1986, a second Megatek was returned to the central computing facility because one terminal was found sufficient for 80 engineers. A Xerox 6500 color copier was also returned to the central facility as experience showed that color has application only in select situations.

The local facility communicates with a number of other computer systems within the company, as shown in Fig. 2. The RJE connections to the Sperry 1100 scientific computer system and the CADAM CAD/CAM system use 9600-BPS leased common carrier lines. Five HPC2623A terminals with Tektronix 4631/4632 copy units and four IBM 3251 terminals with two IBM 3255 controllers are available for use with the Sperry and CADAM system, respectively. We are also a part of a company-wide engineering network, a large DECNET network including over 100 VAX nodes, a local CRAY-1S Supercomputer, and a CRAY XMP that is located at Rye Canyon, California. Our network links include a local Ethernet within our own building and a 256-KBPS private cable that connects us to the central Information Services facility in another part of the plant.

Software Development

The primary goals of the new software system were to reduce software development costs and analysis turnaround time significantly. With the initiation of a new software system, the opportunity to examine prior deficiencies and alternative solution approaches was apparent. The evaluation of the project requirements indicated that a consistent set of problems existed, all of which impeded productivity. These problems were categorized as either system or applications software problems and are summarized as follows:

1) Application software developed over the years had experienced many enhancements implemented by different programmers. These continual modifications led to what was referred to as "spaghetti" coding, which was difficult to enhance and maintain and which led to the obsolescence of software when the primary developers left the area. The advent of the new distributed processing hardware system required that the application codes operate on both the VAX and CRAY machines. The older software was difficult to convert due to its unstructured architecture and because specialized machine-dependent coding techniques had evolved over the years.

2) Many different data structures and formats were used within the different application codes. This was perceived as a problem in that the duplication of effort by code developers was taking place but more significantly because the transfer of data among codes led to the development of many data con-

Fig. 1 Requirements.

CODE DEVELOPER / USER CONVENIENCES	HARDWARE REQUIREMENTS								SOFTWARE REQUIREMENTS						
	SPECIAL PURPOSE SCOPES	DESK TERMINALS	MINI COMPUTER	COLOR SCOPES	VECTOR SCOPES	PRINTER/ PLOTTER	SUPER COMPUTER	MATRIX CAMERA	COMMAND LANGUAGE	DATA BASE STANDARDS	GLOBAL DATA DATA MANAGER	CADAM	SPECIAL APPLICATION CODES	LOCAL DATA BASE MANAGER	STANDARD GRAPHICS
ACCOMMODATE SOFTWARE DEVELOPMENT		√	√	√	√	√			√	√	√			√	√
INTERACTIVE ANALYSES	√	√	√			√			√	√	√		√	√	
INTERACTIVE MESH GENERATION	√	√	√		√	√		√	√	√	√	√	√	√	√
INTERACTIVE MASS PROPERTIES	√		√			√						√			
LOCAL PRE PROCESSING		√	√						√	√	√		√		
DRAWING RELEASE SCHEDULE		√	√			√				√			√		
STATUS OF ANALYSES		√	√										√		
TEST ANALYSES		√	√			√			√	√	√			√	√
LOCAL POST PROCESSING	√	√	√	√	√	√		√		√			√	√	√
EFFICIENT OFFSITE BATCH RUNS		√	√			√	√			√	√		√		

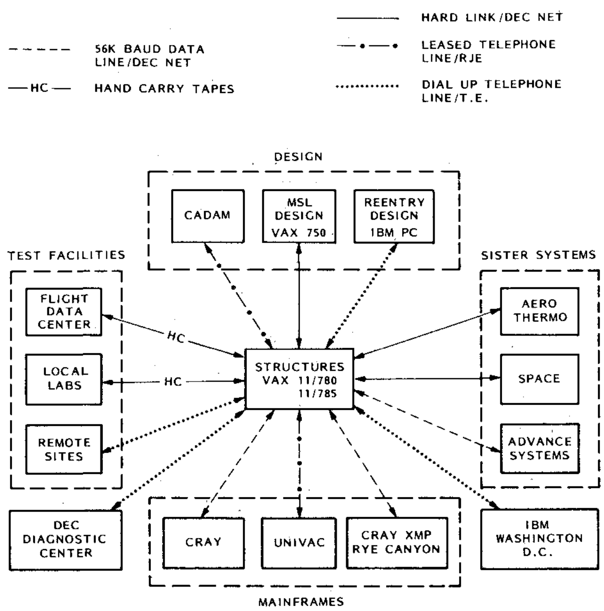


Fig. 2 Data links.

verters that were typically undocumented, redundant, and had unique applications. This problem became exacerbated when transferring data files from machine to machine.

3) There were no standard user I/O interfaces for the application codes. Each code used different free-field data entry techniques and again prompted the duplication of software. This also required users to learn several different code I/O systems along with the idiosyncrasies of each.

4) Many analysis tasks required repetitive executions to perform parametric design studies. Analysts performed these tasks by editing runstreams with slight changes and learning how to best utilize the techniques peculiar to each code. A general system did not exist for automating repetitive computational analysis and data reduction tasks.

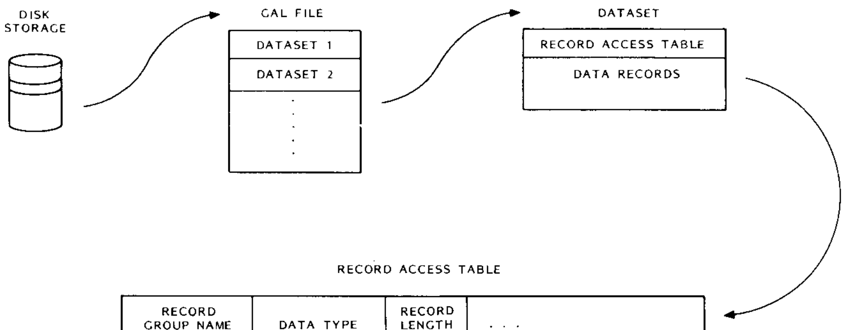
5) Much of the analyst's time was spent in performing the specialized postprocessing of computed data that varied from day to day. Analysts were spending significant time generating ad hoc FORTRAN programs for manipulating and displaying data.

Having identified these problem areas, it was quite apparent that significant improvements could be made on the new software system. The major thrust of the improvements was to establish software coding standards, standard data formats with a common global database manager, a free-field interpretive language with UNIX-like capabilities that allow procedures for program flow control, and the development of new application codes to perform structural analyses with very general data display capabilities. It was agreed that the cited standards would be used for all new applications software development and would be retrofit into existing software as appropriate.

System Software

The general standards adopted for the development of the new software system are described in Ref 1. These include the adoption of the ANSI X3.9-1978 FORTRAN 77 language[3] and Master Source Code (MSC) programming techniques to improve code modularity, simplify code architecture, and isolate machine dependencies. The MSC technique requires writing FORTRAN in a prescribed format that includes blocks of code which are segregated as comment statements but include the required key words and data describing the software. This technique allows the inclusion and isolation of machine-dependent portions within one version of the software. As such, the coding is not a directly compilable FORTRAN source code but must be operated upon by an applications program that forms the compilable FORTRAN version. The FORTRAN 77 and MSC requirements have already proven to be invaluable in certain applications. When creating software, the developers have programmed for operation on several machines simultaneously using MSC, thereby simplifying the conversions. The modular programming techniques developed using FORTRAN 77 have demonstrated their effectiveness in nonlinear analysis codes where certain structural elements have easily been entirely removed and replaced with different fomulations within hours.

The CLIP user interface package[4] was adopted for all new software development. At its lowest level, CLIP is a package of FORTRAN routines that serve as a free-field input reader. It was adopted to provide uniformity of program input syntax and to decrease coding duplication. By adhering to this standard, each new code developed will now be similar from the user's vantage. In addition to serving as a very versatile free-field reader, CLIP provides facilities for accessing separate documentation or HELP files during the execution of the host processor to facilitate the use of interactive codes. At its highest level, CLIP permits the user to form command procedures with variable names as input parameters to the host code and allows for looping and branching within runstreams. These command procedures also allow the execution of independent codes that may interact with one another within a runstream to perform new tasks not forseen by developers. These techniques have proven very effective for conducting parametric studies and in performing statistical analyses using



RECORD ACCESS TABLE

RECORD GROUP NAME	DATA TYPE	RECORD LENGTH	. . .
STANDARD	CHARACTER	3	
CONTENTS	CHARACTER	40	
OWNER	CHARACTER	20	
UNITS	CHARACTER	32	
ABSTRACT	CHARACTER	1000	
TIME	REAL	800	
DISPL	REAL	800	
VELOCITY	REAL	800	
ACCEL	REAL	800	

Fig. 3 Nominal data structure.

Monte Carlo techniques without modifications to the application codes

The problems cited with nonstandard data formats were addressed by the selection of a currently available database structure, or Global Access Libraries (GAL). This selection was also predicated on the availability of the well-documented EZ-GAL[5] global database manager system. This package is a collection of FORTRAN routines for the input, output, and manipulation of data files. EZ-GAL is an excellent vehicle for maintaining a central, self-descriptive global database. The utilization of this package avoids software duplication and provides consistency in the global database structure. With the selection of the global database manager, the logical format of the data structures needed to be defined. The development of standard formats has been a formidable task, since generality is needed to allow the storage of many forms of engineering data and similarity of formats is needed to allow operations by a single data manager. Early attempts resulted in positional data structures wherein records are identified by their position. These attempts required numerical values to determine the nature of the data and indexing records to show the logical structure of the data. This approach was governed by the limitations of the original data manager and was abandoned after several months due to the complexities of its implementation by developers and because of its obvious deficiencies in directly describing the engineering significance of the data.

A new format was developed that allowed character and numerical data to be mixed within the same data structure. The new format identifies data records by names rather than position and is called a "nominal" data structure. EZ-GAL was modified to perform the necessary operations for the nominal datasets and is now in full operation. The nominal data structures allow for the complete description of the information contained within a dataset. Figure 3 presents a simplified display of the nominal data structure. As shown, disk storage consists of global access libraries or GAL files, each of which consist of a collection of datasets. Each dataset contains the data records and a record access table that contains the self-descriptive information for the dataset. The "standard" and "contents" records are mandatory for all datasets. The standard record must contain a prescribed character string that identifies the dataset as being a member of the standard format system. The contents record must be included and contains a character string name that is assigned for a particular data format such as DOF for degree-of-freedom tables or nodes for nodal coordinates. The particular format shown in Fig. 3 is for physical transient response quantities, and thus the contents record would contain the character string "physical-response/time/transient." The records containing the numerical data are identified by names with engineering significance as shown. I/O functions are performed by EZ-GAL through subroutine calls that identify the records by their engineering names in the record access table. Optional records, which may include a narrative describing the data, may also be included to further describe the data, e.g. owner, units and abstract. This has already proven useful for the storage of test data where the abstract record has been used to describe the test procedure and instrumentation locations. Twenty different data formats have been developed to date, and a document stored on the VAX computer has been created to describe and catalog each format thoroughly. This document is used by both developers and analysts and contains formats ranging from finite-element modeling data, applied loads, and stress and strain results to modal data and dynamic response quantities.

The local database, as opposed to the global database, is highly volatile and is used by the host program. The key to effective local data management is computational efficiency. The proper selection of the local database manager is somewhat dependent on the application requirements of the processor and computer storage attributes. Programmers have developed their own local database managers as appropriate for each application code. However, with the cooperative effort among developers that has been achieved during development of the new integrated system, very few unique local database managers are now in use.

Applications Software

As expected, both general and specific codes have been developed to analyze the unique events associated with the product. The architecture of the integrated system of processors around the global database with self-describing data formats is shown in Fig. 4. The integrated system is now totally functional and that with the established architecture, the installation of any new applications software is easily accommodated. Operational codes are used for assessing structural responses to underwater depth charge shock, launcher in-tube dynamics, and closure vehicle dynamics. Other specific codes, having broader applications, include DISCO-II for 6-degree-of-freedom dynamic analyses of separating bodies, NEWTON for transient, steady-state, and random, modal response analysis with a modal displacement fomulation, GALPRO a general data analysis graphics package with a wide range of spectral analyses, and MSLOAD for internal load distribution analysis using modal acceleration formulation. DIAL is the general code used for the conceptual and final design of flight hardware and ground support equipment. Special modeling capabilities have allowed the analysis of the vehicle roll stability of an 8-axle truck with isostatic suspension. DIAL is a company-sponsored code (available for external distribution) that has been tailored for the design and analysis of flight vehicles.

As shown in Fig. 5, the hardware/sofware enhancements have allowed the engineer to reduce the analysis cycle time to a span consistent with the needs of rapid design support. Most of the technology has been available since 1978, and a three-dimensional solid model of a motor with 35,000 degrees-of-freedom was analyzed then with the Texas Instrument Advanced Scientific Computer. This exercise disclosed the need for the effective transfer of large blocks of data, automatic mesh generation and debugging, efficient algorithms for numerical computations, the effective post-processing of large data blocks, and the checkout and debugging of the DIAL code. Some key attributes leading to the reduction of analysis time cycles include the acquisition of the CRAY computers with private cable lines and software enhancements such as a state-of-the-art Lanczos eigenvalue package, state-of-the-art preprocessing, a unique, integrated local database and database manager for rapid postprocessing, and a wide variety of special features that facilitate the analyses of solid rocket motors. A brief description of the key application codes follows.

The DIAL finite-element system[6] is a general-purpose, two- and three-dimensional finte-element analysis system consisting of a family of independently executable modules. These modules communicate with one another through a common database that resides on disk storage (Fig. 6). DIAL can perform both linear and nonlinear static and transient structural analyses, extract frequencies and mode shapes for vibration analysis, and perform bifurcation buckling analyses. The bifurcation buckling and vibration analyses can be peformed from either a linear or a nonlinear prestress state. Within the realm of linear analysis, the response of axisymmetric structures to nonsymmetric loadings can be calculated using the Fourier decomposition method. The assumed linearity allows the Fourier decomposition method to be used for vibration analysis and bifurcation buckling as well. Other analysis capabilities include fracture mechanics, system vector and matrix manipulation, general Rayleigh-Ritz transformations and system reduction (substructuring), linear and nonlinear steady-state heat transfer analysis, linear and nonlinear transient heat transfer analysis, sequentially staged thermal-structural analysis, and other field analysis problems such as diffusion and electrostatic potential.

The element library consists of a full set of modern, multiorder isoparametric solid elements along with the corresponding degenerate curved shell and plate elements. Gap and contact elements, constraint elements, straight and curved beam elements, lumped nonlinear springs and dashpot elements, crack tip elements, and two-dimensional approximate star grain slot elements are also included.

The material library consists of general anisotropic linear materials along with isotropic elastic-plastic, nonlinear elastic orthotropic, isotropic incompressible, isotropic high-temperature creep, and incompressible hyperelastic materials. Nonlinear contact and friction can also be modeled. In addition, multilayer composite shell configurations can be modeled using laminate theory. All materials are considered to be temperature-dependent.

Extensive pre- and postprocessing capabilities have been incorporated into DIAL. Advanced automatic mesh generation capabilities greatly facilitate model preparation. Extensive

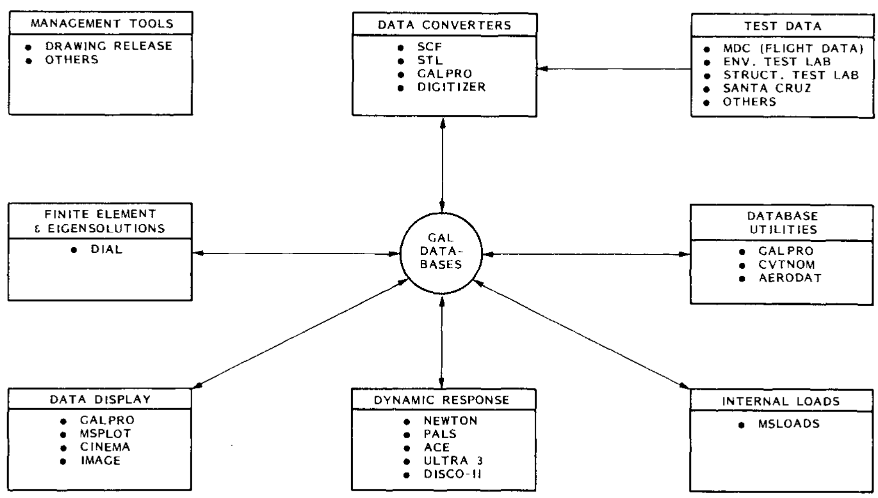


Fig. 4 System architecture.

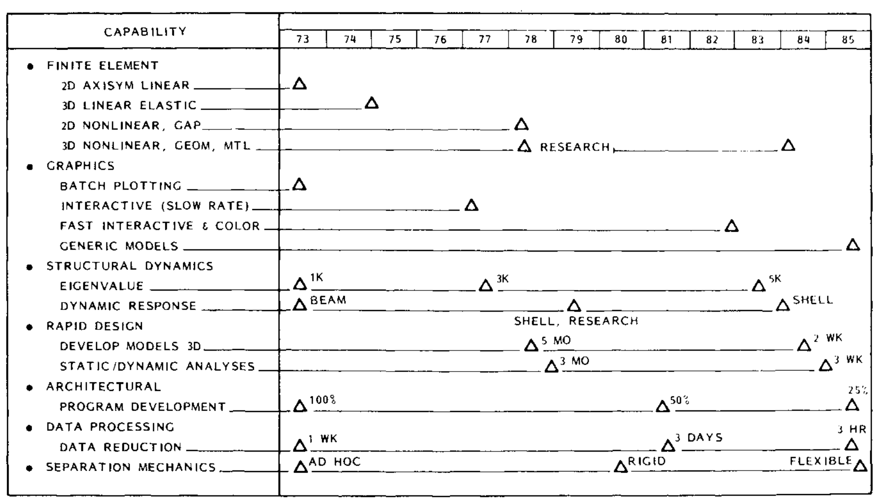


Fig. 5 Structural analyses.

model plotting capabilities available on a wide variety of graphics devices help to facilitate model checking. Undeformed shape model plots with node and element labels, hidden line and surface plots, element shrink plots, and light source shading plots are all available. Real-time hardware rotations and translations and color imaging are available for model plots. The analysis results can be superimposed and extrapolated from integration points to nodes and can be output in a wide variety of ways, including deformed shape plots, stress contour plots, and time/load histories of nodal or stress quantities. Analysis data can be scanned, highlighted, filtered, and summarized in order to make the data easier for the user to evaluate.

Restart capability is provided at almost any point in the analysis. Because of the modular database architecture, user control is very versatile and economical. Problems can be solved in either interactive or batch modes or through a combination of the two. DIAL is operational on a variety of computers including CRAY, Digital's VAX series, and the Apollo series, and is compatible with the decribed system software.

Two eigensolution algorithms, subspace iteration and Lanczos, are available for extracting eigenvalues in the EIGEN processor of the DIAL system. Both algorithms have been implemented with an automatic shifting capability that balances vector iteration and matrix decomposition execution times. For buckling mode analyses, an interval of interest and/or an initial shift may be specified. For vibration analysis, the program will extract the modes in ascending frequency until either the number of modes requested are found or all modes below a desired frequency are extracted. The generalized mass, modal participation factors, and percent total mass for each mode and direction are calculated for vibration modes. The percent total mass is particularly useful for identifying the type of mode (x, y, or z direction response) and for determining the completeness of the modal set. Either algorithm, subspace iteration or Lanczos, can be used for vibration analysis, but only subspace iteration can be used for buckling analysis. To extract 90 modes from a model of 3444 equations (521-rms bandwidth) using the Lanczos algorithm required 112 secs on the CRAY-1S machine.

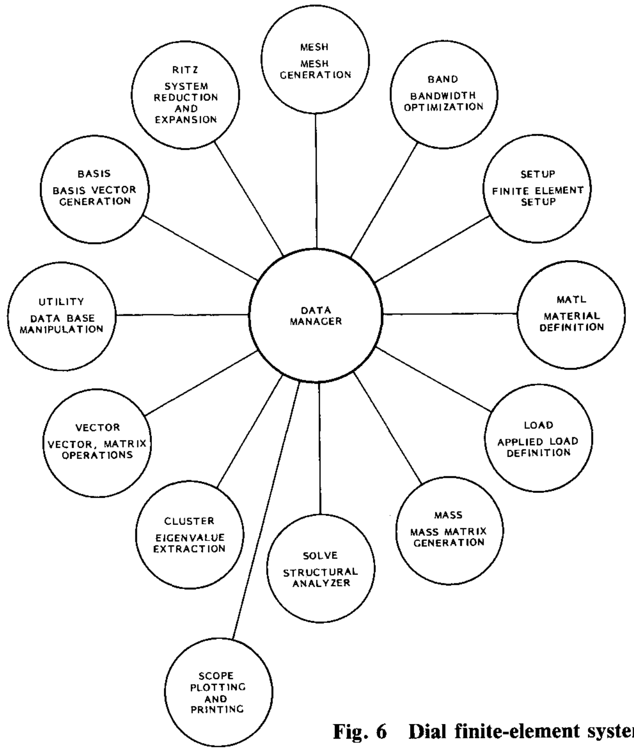


Fig. 6 Dial finite-element system.

DISCO-II[8] is a system of programs for performing general-purpose, 6-degree-of-freedom dynamic analyses of multiple rigid and flexible bodies that interact with their surroundings and with each other. DISCO-II is applicable to a wide variety of dynamic, separation, and deployment problems including vehicle separation, submunition dispensing, and multibody impact events. There are no built-in constraints for bodies being analyzed since each body in the simulation moves only as a result of initial velocities or body forces. There are currently eight different types of body forces considered, including aerodynamic loadings, applied force-time histories, beams and/or dashpots connecting two bodies, gravitational forces, pistons and/or pointer springs connecting two bodies, and point-surface impacts between two bodies. The DISCO-II family of programs includes AERODAT to create aerodynamic coefficients when aerodynamic loadings are required, SETUP to read and check input data before being used by the analysis module, and IMAGE and CINEMA for generating pre- and posthardcopy and motion picture plots showing the configuration being analyzed and results obtained. It is user-friendly, has extensive master/slave body capabilities, and can treat events that are triggered by time and /or by displacement conditions. Its efficient, variable timestep integration algorithm facilitates its use for performing Monte Carlo analyses.

NEWTON is a dynamic response program that performs transient, steady-state, and random response analyses using the modal displacement approach. It is a command-driven interactive program that accepts eigendata and FEM model data from the global database created by DIAL and computes the damped response due to initial conditions, applied forces, or prescribed displacement histories and ground accelerations. Transient forcing functions may be described as piecewise linear or harmonic loadings using Fourier coefficients. Damped steady-state solutions are performed for either Fourier harmonic loadings or frequency-modulated sinusoids. Random response is performed for uncorrelated stationary excitation. NEWTON has dataset manipulation and output plotting capabilities, but most postprocessing is performed with GALPRO.

The lateral shock code PALS computes vehicle loads and dynamic responses as a result of a lateral shock imparted to the vehicle. The shock is transmitted through the suspension system to the vehicle through a pad liner support system designed to support and protect the vehicle. High lateral vehicle velocities relative to the suspension system cause the surrounding gases to act as elastic springs. PALS handles vehicle geometric bow, gravity forces, and applied forces and moments at any node and uses a variable-step time integrator of the Runge-Kutta-Fehlberg type. The vehicle is considered to be an elastic body and is modeled using modes computed with a finite-element code. The pad liner support system is viscoelastic and highly nonlinear, and each pad can be modeled separately. PALS communicates with other codes through the standard global database.

The ULTRA3 code computes vehicle trajectory and loads during the launch event. During upward travel, the vehicle experiences lip seal forces, gasdynamic forces, inertial and friction forces, fluid mechanical forces, forces caused by interaction with the nonlinear, viscoelastic pad support system, and axial thrust caused by high-pressured gas from the launcher. Vehicle models are represented by modal characteristics computed with a finite element code, and the suspension system model is characterized by highly nonlinear elements. ULTRA3 communicates with its pre- and postprocessors through the standard global database.

ACE is a set of (two) programs, extensions of the DISCO-II system's SETUP and DISCO2 programs, for conducting transient response analyses of missile configurations during the "closure event," during which the vehicle breaks through the protective dome structure. ACE utilizes DISCO-II special-purpose entry points to take fluid and gasdynamic phenomena into account and to treat state variables in addition to those required for ordinary DISCO-II models. ACE (like SETUP and DISCO2) uses the CLIP processor for command interpretation and the utilization of command procedures and the EZ-GAL file management system for archiving global data. Postprocessing for ACE is done by GALPRO.

MSLOAD is a separate processor that relieves transient analysis codes such as NEWTON, PALS, ULTRA3, and ACE from the burden of computing internal vehicle loads. MSLOAD receives external forces and generalized or physical displacements from a transient-analysis processor, vibrational modes from an eigensolver, and a factored system stiffness martix and element data from a finite-element processor like DIAL. MSLOAD computes internal load distributions for constrained or unconstrained structures from physical displacements and element stiffness information. MSLOAD is also capable of computing physical displacements from generalized transient response, frequency-based steady-state response, or time-based steady-state response information, using either modal displacement or modal acceleration techniques. This code benefits greatly from the system software since it interacts with several codes and combines static and dynamic solutions as a function of time.

The GALPRO program, a general utility interface that provides a set of comprehensive interactive command operations for all computer codes that produce GAL-formatted data,[7] is the user's window into the global database. This program was conceived to relieve applications codes from the burden of data display and graphics output. Most applications software now only produce output to the global database and depend on GALPRO for postprocessing. GALPRO has a comprehensive set of data display commands that range from print commands that create report-ready tables of data with many format options to graphic displays ranging from simple x-y plots to descriptive three-dimensional data plots with many options for titles, line styles and symbols, and automatic note and legend generation. GALPRO also includes a myriad of general analytical commands including matrix, algebraic, and statistical operations as well as digital signal processing functions such as PSD, FFT, automatic filtering, and shock spectra. GALPRO commands may utilize a special matrix operation language that is used in much the same way as one would write matrix equations in the FORTRAN language. This program has many utility commands used for searching, sorting, copying, and editing data and many commonly used functions such as integration, differentiation, interpolation, and curve fitting.

GALPRO uses the CLIP I/O processor, and, when coupled with the power of CLIP in its advanced form, which allows for looping, branching, and the generation of macroprocedures, it has become and extremely powerful tool. A very simple set of EZ-GAL database I/O FORTRAN subroutines has been developed and distributed to the analysts. When the analyst desires to write any simple FORTRAN program for some very specific application, he can link the output data into the standard data formats and then have access to GALPRO and all its power for subsequent processing.

TEST DATA comes from a variety of sources, such as flight test telemetry, in-house laboratories, and subcontractor test labs. Many different types of data are generated, and each is typically transferred to our computer system for analysis, comparison to model predictions, and processed for reports. Agreements have been reached with several of the data sources on the formats of the data transmissions. As explained in the hardware section, links have been established between our VAX and various labs. We have found that the most effective and economical approach is to allow the labs to dictate the most convenient data format for them and for us to write simple data converters to translate the data to the standard formats. Several converters have been created to date, and more are expected as the testing phase of the missile development program accelerates.

MANAGEMENT TOOLS have been created to manage the engineering design and analysis. A common problem found in analysis support organizations is tracking schedules for design drawing release. One of the design organizations with our division maintains its scheduling information on a VAX/750. The system has been linked to our VAX/780 via DECNET, and software has been developed which automatically queries the VAX/750 on a weekly basis for updates to the schedules. This information is used to update our database, which contains additional data such as personnel assignments, stress analysis summaries, and documentation references unique to us. The system has proved useful in monitoring work assignments and tracking design. A STATUS processor was designed to create a database depository for key engineering information such as mass properties, design loads, and stress margins of safety. Our experience with the STATUS processor has shown that the system is too complex, and simpler approaches have developed by using the VAX/VMS text files and the spreadsheet available for the Rainbow (PC100).

Summary

Our successfully completed plan for the integration and improvement of analysis support capabilities has had significant impacts on design. This became evident in a recent, eight-day program review that was summarized by a former chief engineer who commented that "The analytical models are much more detailed than they were at this point in the previous development program and they are interacting with the subsystem design organizations better than ever before. There is strong evidence that they are having a good influence on the present design and not simply reacting to the designs produced by the subsystems."

Acknowledgments

At the risk of not recognizing the efforts of other persons who may have also contributed significantly, we wish to acknowledge Messrs. René Bousquet, Robert Clark and David Cunningham, Drs. Norman Cyr and John Dickens, Mr. Laurence Edington, Dr. Carlos Felippa, Messrs. Gordon Ferguson, Laurence Langmead, William Loden, Michael Manke, Michael Murphy, Jan Schipmolder and Brian Stocks, Drs. Lyle Swenson and Frank Weiler, Ms. Mary Wright, Dr. Richard Wu, and Mr. Jeffrey Wurtz for their fine technical efforts, Ms. Debra Butler and Ms. Kelly Dickson for their assistance in the preparation of this report, and Dr. Soloman Singer for providing continued program guidance and resource support throughout this project.

References

[1]Wong, D.G., Bousquet, R.D., and Fuller, C.R., "Integrated Structural Analysis and Design Support for Advanced Launch Vehicles," AIAA Paper 82-0675, 1983.

[2]Weaver, L.K., "CADAM Description and Application," Lockheed California Co., Burbank, CA, LR 27362, 1976.

[3]American National Standards Programming Language FORTRAN, ANSI X3.9-1978, American National Standards Institute, 1978.

[4]Felippa, C.A., "A Command Language for Applied Mechanics Processors," Lockheed Missiles and Space Co., Sunnyvale, CA, LMSC Report D878511, 1981, Rev. 1985.

[5]Felippa, C.A., "The Global Database Manager—EZ-GAL," Lockheed Missiles and Space Co., Sunnyvale, CA, LMSC Report D766995, 1981.

[6]Ferguson, G.H. and Cyr, N.A., "DIAL Structural Analysis System, Revision L3D1," LMSC Report, March 1986.

[7]Loden, W.A. and Stocks, B.B., "Global Access Library Utility Programs," LMSC Report D850140, Aug. 1986

[8]Swenson, L.W. Jr. and Wurtz, J.D., "DISCO-II—Dynamic Interaction Simulation of Clustered Ordnance," LMSC Report D850147, May 1986.