Craig S Collins

MANAGING ENGINEERING DATA: EMERGING ISSUES

edited by
R. E. FULTON

MANAGING ENGINEERING DATA: EMERGING ISSUES

presented at

THE 1988 ASME INTERNATIONAL
COMPUTERS IN ENGINEERING
CONFERENCE AND EXHIBITION
JULY 31–AUGUST 4, 1988
SAN FRANCISCO, CALIFORNIA

sponsored by

ASME/COE DATA BASE COMMITTEE
COMPUTERS IN ENGINEERING DIVISION, ASME

edited by

R. E. FULTON
GEORGIA INSTITUTE OF TECHNOLOGY

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
United Engineering Center 345 East 47th Street New York, N.Y. 10017

A FINITE ELEMENT MODEL AND FINITE ELEMENT ANALYSIS RESULTS RELATIONAL DATABASE MANAGEMENT SYSTEM

C. S. Collier
LTV Aircraft Products Group
Dallas, Texas

W. J. Rasdorf
Department of Civil Engineering
North Carolina State University
Raleigh, North Carolina

1 ABSTRACT

A Finite Element Model (FEM) and Finite Element Analysis (FEA) Results Relational Database Management System (RDBMS) has been developed at LTV Aircraft Products Group to provide our engineers with a new capability and a new tool to help them be more productive.

The amount of FEM data required to represent an aircraft and the volume of generated FEA results is enormous. Model sizes have been as large as fifty thousand lines, and output up to ten million lines. The maintenance of the model (input) and the interpretation of the FEA results (output) are labor intensive, time consuming tasks. In fact, the jobs of making and verifying the FEM, as well as "pulling loads" from the results, are major milestones on the master engineering schedule.

The FEM-FEA RDBMS provides a way to store MSC/NASTRAN^{TM} FEM input data together with the solution output while providing a means for quick interactive retrieval of the data. On a computer terminal, the user may easily display the data using any criteria he/she wishes to specify. The FEM-FEA RDBMS assists engineers with locating worst case loads and associating them directly with the model input data and structure.

This paper defines the FEM-FEA RDBMS application in detail and identifies three distinct user interfaces provided, two of which use SQL, the national standard query language for RDBMs accepted by ANSI in 1986. Illustrative examples are presented to demonstrate the benefits of using a RDBMS for FEA results interrogation and FEM maintenance.

2 INTRODUCTION

The primary loads on an aircraft are pressure flight loads on the upper and lower surfaces and concentrated ground loads on the landing gear. These 'external' loads distribute throughout the aircraft structure (load paths) to cause the structure to be in equilibrium. Finite element analysis programs are used to map out load paths and to compute the forces on each individual component of the aircraft. First though, a finite element model (FEM) must be made of the aircraft structure. Then all of the applicable load conditions are applied to the model.

In addition to the primary aircraft loads, there are many other load conditions which must be considered: flight maneuvers, takeoff and landing, environmental effects such as gust and temperature variations, special conditions such as engine failure and tire blow out, etc. In addition to evaluating the effects of each individual load condition, combinations of load conditions must also be evaluated.

2.1 Analysis of Aircraft Structures

From all of the external load conditions mentioned above, the worst computed internal load per structural component must be found to enable the engineer to analyze the structure and then to design it by sizing its individual components. Based on a structural component's configuration, size, and material composition, the engineer is able to determine its allowable load. A margin-of-safety (allowable load divided by the actual load minus one) is then calculated. Any structural component which has a negative margin-of-safety must be redesigned.

2.2 Finite Element Models

A FEM is an analytical representation of a real structure. As shown in Figure 1, FEMs are used to represent the general stiffness of an aircraft. The webs of spars, ribs, and bulkheads, as well as the fuselage and wing surfaces, are modeled with shell type elements (planar elements which have 3 or more grids). A grid defines an X, Y, and Z point in space. Fuselage stringers, spar, rib and bulkhead caps, and stiffeners are modeled with beam type elements (line elements which have 2 grids).

MSC/NASTRAN^{TM} FEM data can be categorized into four basic levels as shown in the four tables of Figure 3. The first level represents a single data type, the GRID. The second level represents the elements. Examples in this paper refer to only two kinds of elements: a shell element called CQUAD4 and a beam element called CBAR. Physical properties of elements are represented at the third level. All shell elements, for example, have a thickness; beam elements have cross sectional areas, moments of inertia, etc. assigned to them. The fourth level is the material property data. Shell and beam elements have modulus of elasticities, Poisson's ratios, etc. assigned to them at this level.

Referring to Figure 1, the skin and web element thicknesses are included with shell property data. Stringer, cap, and stiffener element cross sectional areas and moments of inertia are included with beam property data. In turn, the entire aircraft structure's shell and beam element modulus of elasticities and Poisson's ratios are included with material property data.

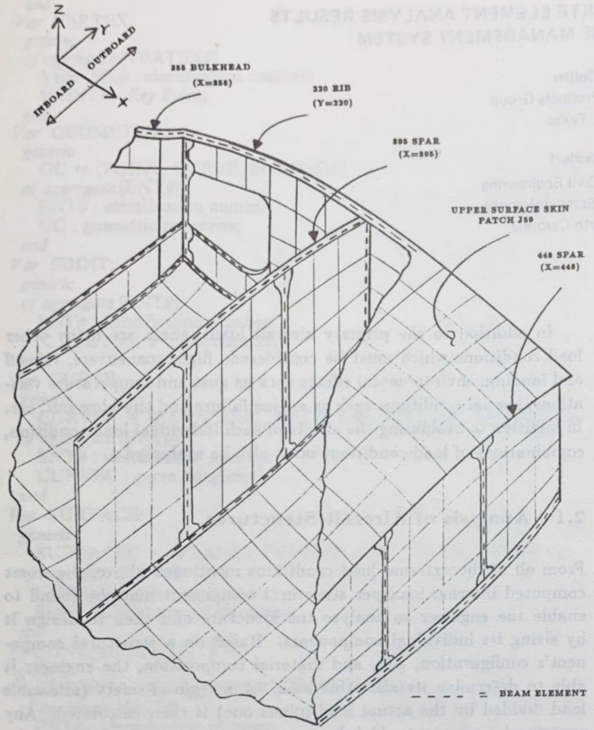


Figure 1: Part of an Aircraft Structure and its FEM

2.3 The Problem

A structure is modeled using many beam and shell finite elements. The model can be visualized as rows of data, each of which represents either a grid, element, property, or material. The number of grids and elements used to represent a structure varies. However, every model needs grids at intersections of major structural components. A skin panel usually consists of many elements called a mesh. As more elements are used in the model the mesh becomes finer. Note that a real structure, having an infinite number of points, is nonetheless modeled with a finite number of points and elements. Therefore, the finer the model mesh the better the analytical results, but also the greater the work load for the engineer and the computer. The problem of effectively balancing solution accuracy requirements with model size is the engineer's challenge.

The balance between solution accuracy and model size almost always leaves solution accuracy as the winner. As computers have become faster and less expensive, the tendency has been, and will continue to be, to build larger models. However, the larger the model, the longer it takes to build, verify and run, and to interpret its results. Compounded with larger model sizes is the additional burden of interpreting the results of the many external load conditions which are applied to the models.

The problem of maintaining FEM models and interpreting their massive amount of output has not been properly handled with conventional procedural languages. A solution is to store the large volume of FEM-FEA input and output data into a RDBMS. Such an approach is described herein.

2.4 What Is a RDBMS?

A relational database management system provides powerful data storage and retrieval capabilities using an English-like query language. With the language, specific data may be retrieved using clauses like '*where displacement in GRID > .01*' and formatted using clauses like '*sorted by displacement.*' The data itself is stored in 2-D tables. Although they may seem unrelated to each other, the software is able to link tables together when the database developer defines keys (unique data identifiers) for each table and refers to them in other tables.

3 THE FEM-FEA RDBMS APPLICATION

The following sections of the paper describe in detail the FEM and FEA data stored in the FEM-FEA database and a mechanism for linking the data together. Figure 2 represents the flow of FEM input and FEA output data into the database. Note that external loads and FEM data are both necessary to obtain a FEA solution.

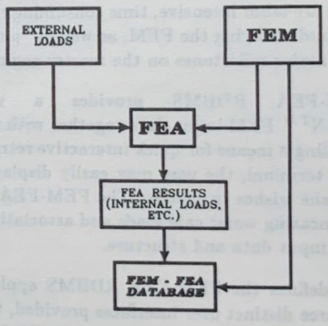


Figure 2: Flow of Data Into the FEM-FEA RDBMS

3.1 FEM Tables

Each different type of finite element input data line is stored in a different table. The most frequently used data tables are: the nodal point GRID; the shell elements CQUAD4 and CTRIA3 and their corresponding shell property data PSHELL; the beam element CBAR and its corresponding beam property data PBAR; and the material property data MAT1.

For illustrative purposes, four data tables and their key links, which are a subset of those introduced above, are shown in Figure 3. These tables represent major data type categories. They have been drawn from the larger FEM- FEA RDBMS we have developed.

The first table, GRID, contains the X, Y, and Z coordinates, noted as X1, X2, and X3, of each grid. The second table, CQUAD4, contains the four grid ID's noted as G1, G2, G3, and G4 that define the corners of a QUAD element. The third table, PSHELL, stores the QUAD element thickness, bending stiffness parameter, and transverse shear thickness of the QUAD elements noted as T, 12I/T3, TS/T respectively. The fourth table, MAT1, contains the modulus of elasticity, shear modulus, Poisson's ratio, etc. of the QUAD elements noted as E, G, and NU.

Each data table has a unique ID. For grids it is GID, for elements EID, for properties PID, and for materials MID. Because of these ID's, a row of data from one table can be related to a row of data from another table. The developer and user of the application will know these relationships and will use them to link data together. For instance, the PID on the element data line allows the element connectivity data to be associated with the element properties.

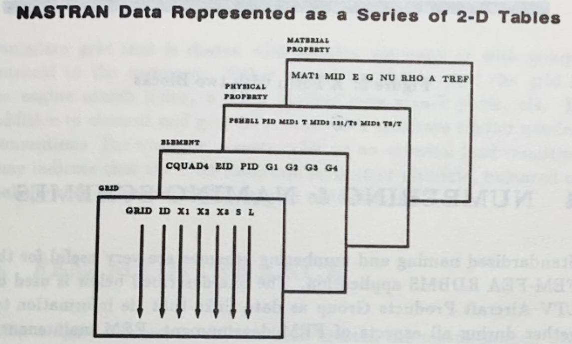


Figure 3: NASTRAN Data Represented as a Series of 2-D Tables

3.2 Linking FEM Tables

Once the linking ID columns between tables have been identified, the RDBMS software can pull the data together and display the information based upon any 'where' clauses provided, i.e., in a variety of combinations to suit the user's needs. One method for linking tables together that we have found to be quite useful is to create views. A view is a pseudo table that can be created by combining portions of many existing tables. A view does not contain data, thus, redundant data is not generated. Nevertheless, views have all the functionality of real tables. Queries can be performed on them using the same syntax.

Relationships between tables can be classified in one of three ways: one-to-one, one-to-many, and many-to-many. Ironically, the most basic finite element data relationship, grid-element, is a many-to-many relationship, which is the most difficult to work with. Each grid can be referenced by many elements and, depending on the element, each element references two to four grids.

To provide the necessary database links between grids and elements requires the creation of a linking table. The new table contains two linking columns: the GID from the grid table and EID from the element tables. There is a one-to-many relationship between the new table and the other two tables. One-to-many and one-to-one table relationships are easy to work with. Most of the FEM-FEA table relationships are one-to-many. One such example is the relationship between a property table and an element table. Each element record references one property record but each property record can be used by many element records.

3.3 Linking FEM and FEA Tables

By linking the FEM input data tables with the FEA output data tables the engineer has a powerful way to simultaneously see FEM-FEA data together. As a result, the engineer has acquired a previously unavailable capability to rigorously and selectively query his database. Queries can be made to display any separate pieces of information together in any format. Information as complex as *display load conditions and all elements in descending order that are on the 355 bulkhead, are less than .4" thick, are made from 7075 aluminum alloy, have more than a 2000# shear load in them during a gust load condition, and displace in the out-of-plane direction by more than .033"* is attainable.

3.4 The FEM-FEA RDBMS Cycle

Two steps that occur during the FEM-FEA RDBMS cycle for analyzing a particular structure are:

1. Create the structure of the database. That is, define the tables of the database, the columns in each table, and the attribute of each column. In the FEM-FEA application, column attributes are defined as either numbers or characters.
2. Load the data into the tables. Actually read the FEM input data and the FEA output results, segregating like record types and insert them into the appropriate database tables.

Utilities have been written to perform these steps and are included as options in the FEM-FEA main menu shown in Figure 4.

An engineer may have his own database, or he may share a database with other engineers. Most of the time a group of engineers will pull loads from the same FEA results data, in which case the protection schemes provided with an RDBMS are conveniently used to only allow read access privileges to the data, thus prohibiting unauthorized deletions or changes.

3.5 System User Interfaces

Three distinct levels of interfaces are available with the FEM-FEA RDBMS developed at LTV Aircraft Products Group. Depending on the task at hand, the engineer can either use the query language, use the report command files, or query forms. These interfaces are accessed from the main menu shown in Figure 4. The main menu and application control environment was written in DEC control language, DCL, and runs on a VAX^{TM} 8650 computer.

One interface is the query language SQL (pronounced "sequel"). SQL is a full-featured data definition, data modification, and data query language which allows the user to define the structure of the database, to load and edit data, to retrieve and format the data, and to generate reports. Commands can be entered interactively on-line from the keyboard. The RDBMS executes the commands immediately.

In 1986 the American National Standards Institute (ANSI) accepted SQL as the national standard query language for relational databases. The language is often referred to as a fourth-generation language. It is easier to learn than high level languages, however, its

```
            NASTRAN
LTV      FEM - FEA - DBMS

            MAIN MENU

CL   Create & load a database
AD   Add data to a database
QL   SQL query language
SF   SQL*FORMS menu
FM   Forms menu
PR   Print standard reports
G    Graphs
EF   Write database FEM to external file
DD   Delete data and remove database
H    Help

EX   Exit

Enter selection and press return___
```

Figure 4: The FEM-FEA RDBMS Main Menu

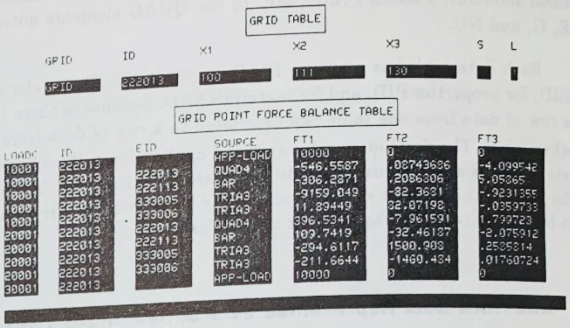
use does require a limited amount of training and a knowledge of the database structure. The benefit to the user is total flexibility of access to his data. He is able to satisfy unanticipated queries "on-the-fly."

Another interface provided to the engineer is a menu of various data reports that he commonly would produce during the FEM-FEA cycle. This interface is the easiest one to use. After selecting a menu option, the engineer responds to a few prompts and a standard report is displayed. Each menu selection represents a specific report format, therefore little data querying flexibility is needed when using this interface. To perform its specific task each menu option invokes a file containing a series of SQL statements. This file is referred to as a database application program. Usually the statements create views from many tables, group data by related categories, and perform queries and subqueries that often times result in complex manipulations of the data. These database application programs are written by an engineer having a substantial knowledge of the database, SQL, and the RDBMS product.

The last interface provided to the engineer is a menu of prepared forms built with SQL*FORMS, an ORACLE^{TM} product. Each form provides direct user access to the data contained in one or more tables. Retrieved data is displayed into the empty blanks on the form. Querying with forms is as easy as filling in blanks. The cursor is placed in the blank(s) of interest and the number or text string to search is entered. Wildcard characters can be used in the blanks.

Figure 5 illustrates a form that displays FEM input data from the GRID table along with FEA results data from the GPFO table. GPFO is an acronym for grid point force balance. Grouped by grid and load condition, this output form shows the three components of force contributed by all elements attached to grid 222013.

By definition, a form constitutes everything shown on the screen. The form shown in Figure 5 is comprised of two blocks. The top block is drawn from the GRID table. The bottom block is drawn from the GPFO table. Note that the form displays one row of data from the GRID table and all associated GPFO table rows (ordered by load condition and EID) that have the same GRID ID; thereby illustrating the one-to-many relationship between the GRID and GPFO tables. These two blocks are 'coordinated', i.e., an engineer can query any blank from either block/table and a trigger will cause all associated data from both tables to be displayed.



GRID TABLE

GRID	ID	X1	X2	X3	S	L
GRID	222013	100	111	130		

GRID POINT FORCE BALANCE TABLE

LOADC	ID	EID	SOURCE	FT1	FT2	FT3
10001	222013		APP-LOAD	10000	0	0
10001	222013	222013	QUAD4	-546.5587	.08743686	-4.099542
10001	222013	222113	BAR	-306.2871	.2086806	5.05865
10001	222013	333005	TRIA3	-9159.049	-82.3681	-.9231355
10001	222013	333006	TRIA3	11.89449	82.07192	-.0359733
20001	222013	222013	QUAD4	396.5341	-7.961591	1.799723
20001	222013	222113	BAR	109.7419	-32.46187	-2.075912
20001	222013	333005	TRIA3	-294.6117	1500.908	.2585814
20001	222013	333006	TRIA3	-211.6644	-1460.484	.01760724
30001	222013		APP-LOAD	10000	0	0

Figure 5: A Form with two Blocks

4 NUMBERING & NAMING SCHEMES

Standardized naming and numbering schemes are very useful for the FEM-FEA RDBMS application. The one described below is used by LTV Aircraft Products Group as data links that tie information together during all aspects of FEM development, FEM maintenance, FEM-FEA interrogation, and data sharing with other engineering databases. In this paper, the naming and numbering scheme is described as it relates to FEM maintenance and FEM-FEA interrogation. It will be shown to provide the engineer efficient access to the data using terms that are most familiar to him.

4.1 Assignment of Structure Name to ID Range

One powerful technique for retrieving specific data from a database is to make use of known patterns in naming conventions. For instance, an engineer can display GRIDs that are on the 355 bulkhead by issuing the query '*select GRIDs where ID like 478%.*' In this case the engineer knew ahead of time that all grids on the 355 bulkhead have IDs that begin with '478.' In general though, an engineer is not likely to be familiar with the model numbering scheme. For this reason, a FEM-structure table was created to directly associate structure names to a range of FEM IDs. A portion of this table is shown in Figure 6. Using the new table, an engineer can issue the query '*select GRIDs where structure name is 355 bulkhead*' to obtain the same data as required above. To further limit the amount of data to display the engineer could request data for the 355 bulkhead outboard. Note that since the FEM-structure name association is saved in a table, changes to a naming convention are easy to incorporate.

4.2 Existing Numbering Conventions

Many FEM numbering scheme conventions already exist. For example, integer values in specified fields in a GRID table can identify the grid as an upper or lower surface grid, and even identify which surface file it belongs to. Values in other fields can indicate that the grid is a

ID_RANGE

STR_NAME	ID_LOW	ID_HIGH
355 bulkhead inboard	478001	478299
355 bulkhead interm	478301	478699
355 bulkhead outboard	478701	478899

ID_RANGE

STR-NAME	ID-LOW	ID-HIGH
355 bulkhead inboard	478001	478299
355 bulkhead interm	478301	478699
355 bulkhead outboard	478701	478899

Figure 6: Structural Component Association to ID Ranges

boundary grid that is shared with another company or with groups internal to the company. Other values can imply that the grid is an engine attach point, a main landing gear attach point, etc. In addition to element and grid ID's, other ID's also have similar naming conventions. For example, a certain ID on an external load condition may indicate that the load condition is limit or ultimate, balanced or unbalanced, a ground, flight, or gust condition, etc.

5 FEM MAINTENANCE

After a finite element model has been created the most common way of maintaining it (making changes and verifying data values) is to use the computer system's editor. To update or even to view data, which is usually stored in various segregated files, requires swapping the files in and out of the editor's buffer. This process is time consuming and costly. In addition, the data manipulation capabilities povided by the editor are in themselves limited.

5.1 The Query Language Interface

Two key considerations emerged relative to a FEM maintenance interface with the FEM-FEA RDBMS: first, that engineers could easily develop a familiarity with a query language such as SQL; and second, that they could readily understand the underlying database structure that was being used to represent their FEM. Therefore, because the two prerequisites for using SQL on the FEM-FEA RDBMS were met, and because maintaining a model required a RDBMS interface that was considerably flexible, the query language was determined to be the best user interface for maintaining FEMs.

5.2 Query Language Examples

To demonstrate how the query language can be used to access and update the database to perform FEM maintenance operations an example is presented here. Suppose the outboard part of the 355 bulkhead has been relocated from fuselage station 355.000 to 355.250. Making this change to the FEM using the query language is easy. The Y coordinate component of grids that belong to the outboard 355 bulkhead would be updated using the query '*update GRID set X2 = 355.250 where X2 = 355.000*.' However, since we do not want to change grids on the entire 355 bulkhead, simply replacing all occurrences of 355.000 in the GRID table with 355.250 would relocate all GRIDS, some of which should not be moved. Instead we make use of the ID_RANGE table shown in Figure 6 and issue the query '*update GRID set X2 = 355.250 where ID between 478701 and 478899*.' If the engineer was not familiar enough with the model to know the naming scheme, then he would first join the GRID table shown in Figure 3 and the ID_RANGE table shown in Figure 6 together and then issue the query '*... where STR_NAME is 355 bulkhead outboard*.' In either case only the outboard bulkhead would be moved and the correct result would thus be achieved.

6 FEM-FEA DATA INTERROGATION

Interrogating data is the most significant application of a database management system for FEMs and FEA. The task of finding the worst combination of loads on a structural member is a time consuming process for stress engineers in the aerospace industry. As described in the introduction to this paper, finite element analysis proceeds by building a model, consisting of many elements, of the entire structure. The model will generally have multiple external loads applied to it. The analysis determines, among other things, forces in the model elements for each load condition. Conventionally computed results of each load condition for the entire model are printed on paper. The problem that this creates is that the volume of results is so great that they do not readily lend themselves to review by an engineer for the purpose of determining the critical load combinations.

6.1 Application Introduction

An aid to the solution of the problem of data interrogation is the query language of the RDBMS, which enables the engineer to interactively view specifically and only the output needed to analyze a part. For this reason, database application programs that contain a series of SQL statements whose purpose is to pull loads from different results output and model input tables and assemble them in the form of a report are used. A collection of these programs has been created for different purposes and made available to the engineers through the menu. Using the menu, the engineer is able to choose which report to generate. Though the files produce standardized reports, the engineer is still able to specify his particular limiting qualifiers that reduce the enormous amount of data to that which he needs.

6.2 XY Plot of Forces

One of the more useful standard reports generated with database application programs includes the use of graphics. One such program manipulates data in such a way as to produce a formatted set of elements and their corresponding maximum and minimum loads. The formatted set is then used as input to a graphics package for plotting load envelopes. Figure 7 is a graph produced this way. The engineer's answers to prompts from the generic load plotting query language file enable him to generate the graph. In this case he generated an XY plot of the inboard/outboard inplane running load (#/in) in the upper and lower skins going across the 220 rib for gust load conditions.

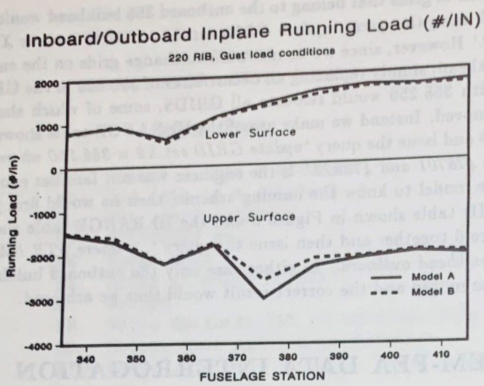


Figure 7: RDBMS Integration with Graphics

6.3 GRID Point Force Balance Report

Another useful standard report that is commonly generated with a file of SQL statements uses output called grid point force balance. This output shows the force on each grid contributed by each element that is attached to it, as well as any applied or constrained forces. This information is used for determining how much load is being transferred across a joint or a structural interface.

Two examples will be used to illustrate the grid point force balance report. The structure used for the examples is the upper beam of the 355 bulkhead, shown in Figure 8 and also shown from an overall perspective in Figure 1. The primary loads are axial, through the bulkhead caps (inboard-outboard direction), and shear through the shear clip (up-down direction). NASTRAN^{TM} grid point force output is used to determine the total load, per load condition, transferred across the rib. Conventionally, the summation of forces and comparison of load magnitudes would have been done by hand and would have taken a long time. Using the query language of the RDBMS to make the appropriate summations and to find the maximum and minimum loads significantly reduces this time. This will be shown in the following axial and shear load examples.

6.3.1 Axial Load

Suppose that an engineer needs to determine, for typical flight loads, the lowest margin-of-safety in the upper beam shown in Figure 8 before the aircraft goes on first flight. The axial force in the upper beam is a major load path, therefore, if the margin is too low, the flight envelope will be restricted accordingly. Before a margin-of-safety can be calculated the engineer needs to know the maximum and minimum (tension and compression) flight manuever inboard/outboard axial loads being transferred through the upper and lower points of the beam (GRIDS 444007 and 444021 respectively). The method of locating the maximum and minimum axial loads is to sum separately the axial force on grids 444007 and 444021 in the inboard/outboard direction (called T1) contributed by all attaching elements that are inboard of the rib (defined as interior on the free body diagram) for each flight manuever load condition. These summed values are then searched for maximums and minimums. The maximums and minimums of these force values

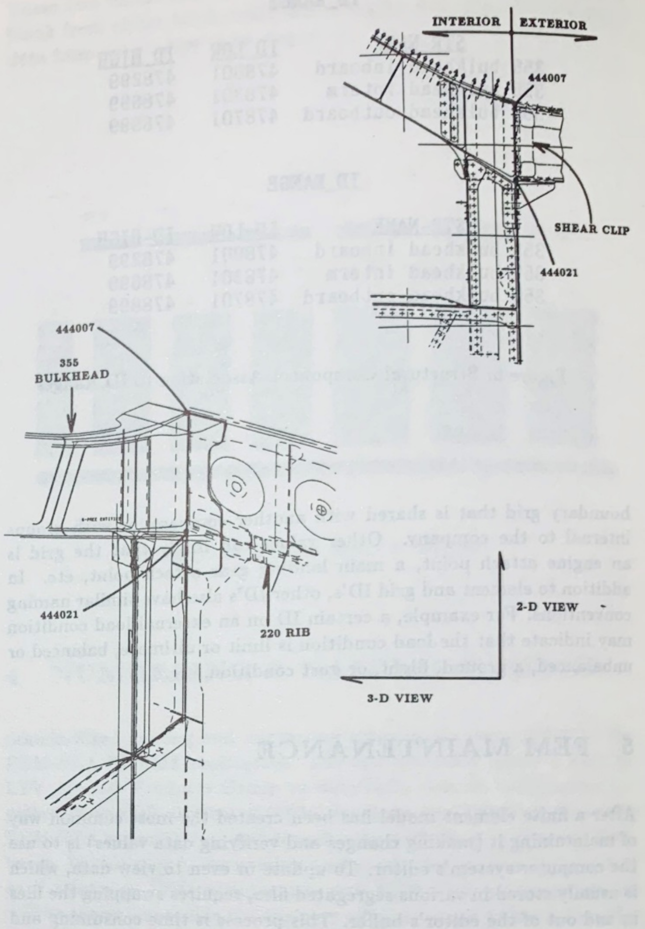


Figure 8: Aircraft Structure, 355 Bulkhead

become the actual loads used, as discussed in the introduction, in the equation: margin-of-safety equals allowable load divided by actual load minus one.

Figure 9 shows prompts from the grid point force balance database application program, the engineer's responses, and the resulting output of loads. The generic application program was able to produce the desired output for the engineer with only five prompts. The real benefit gained by the engineer is that he is able to respond using aircraft structure terminology without having to specify FEM detail.

In Figure 9 the 'type of elements' prompt was left unspecified, in which case all element types are to be included in the solution. If beam had been entered in response to the prompt then only the axial load from the bulkhead caps would be included in the output and not the membrane force of the web and skin. By choosing the 355 bulkhead for elements and the 220 rib for grids, only elements on the 355 bulkhead that attach to grids on the 220 rib (444007 and 444021) are included in the summation of forces. For the last prompt, the engineer has the option of specifying which force direction T1, T2, T3 or which moment direction R1, R2, R3 to output loads.

```
SQL  FILE  PROMPTS          ENGINEERS RESPONSES
load conditions             (flight manuever)
type of element             ( )
element structure           (bulkhead 355)
grid structure              (rib 220)
load direction              (T1)


OUTPUT
Structure        Grid        Load         Load
  Name            ID       Condition    Sum (T1)
BULKHEAD 355    444007       10001      3760.754
                             10021      1434.323
                             10041      2554.980
                             10061      2623.134
                             ....       ....
                             -----      --------
minimum                      10021      1434.323
maximum                      10001      3760.754


Structure        Grid        Load         Load
  Name            ID       Condition    Sum (T1)
BULKHEAD 355    444021       10001      5560.324
                             10021      4633.357
                             10041      2453.008
                             10061      8714.094
                             ....       ....
                             -----      --------
minimum                      10041      2453.008
maximum                      10061      8714.094
```

Figure 9: Axial Load Report

6.3.2 Shear Load

Suppose that an engineer needs to know the total shear force transferred through the upper beam of the 355 bulkhead at the intersection of the 220 rib. The shear force will be used to design the shear clip and fasteners connecting the beam to the rib. The freebody diagram, defined to be the interior area, is the 355 bulkhead, upper skin surface patch J59, and rib 220. Structure outboard of the 220 rib is considered to be exterior to the freebody diagram. The engineer feels that the shear force value from the element force output (computed at the element centroid) is not the true shear at the beam-rib interface. His method of pulling loads for this specific shear force value is to sum together the upward force (called T3) on grids 444007 and 444021 contributed by the interior elements that attach to them, per load condition. The engineer then designs the shear clip to the maximum absolute force value.

Figure 10 shows the four prompts to the engineer that were discussed previously as well as his responses and the resulting output of shear forces. As input the engineer specified his task specific S40_SHEAR table. Since the load conditions prompt was left blank all load conditions are included. Instead of identifying the interior element structure by name, the engineer used a wild card to match a range of elements ID's. All interior portions of the structure whose ID begins with the number 4 are used to satisfy the request.

The database application program that generated Figure 10 was created very quickly by modifying the program that initially generated Figure 9. For shear forces, both of the grids shown in Figure 8 need to be added together for each load condition. Because relational database management systems are non-procedural, they easily handle such unanticipated requests. The solution required the creation of the task specific user table called S40_SHEAR (shown in Figure 11) which associates with structural components the grids that are used for load assignments.

```
SQL  FILE  PROMPTS          ENGINEERS RESPONSES
SC table name               (S40_SHEAR)
load conditions             ( )
type of element             ( )
element structure           ('4%')


OUTPUT
Structure          Load        Shear Load
  Name           Condition      Sum (T3)
S40CLIP1           10001         823.234
                   10021         337.936
                   10041        2554.977
                   10061          23.122
                   ....        .........
                   -----       ---------
minimum            10061          23.122
maximum            10041        2554.977
```

Figure 10: Shear Load Report

```
                S40_SHEAR

SHEAR-PIECE       GRID1       GRID2
S40CLIP1          444007      444021
S40CLIP2          444207      444221
S40CLIP3          444407      444421
```

Figure 11: Structural Component Associated with the Method of Pulling Loads

7 SUMMARY

This paper illustrates FEM-FEA examples from the aerospace industry that demonstrate the usefulness of a relational database management system. Such a system, developed at LTV Aircraft Products Group to aid in finite element model maintenance and results interpretation was described. The ORACLE^{TM} and MSC/NASTRAN^{TM} products were used in the examples shown, as well as in the implementation described. The authors suggest that the advantages gained by combining these two technologies would also occur with other finite element and RDBMS software products.

Representing FEM and FEA data as database tables was explained and the advantages of linking tables together were demonstrated. A naming scheme was introduced relating range of ID's to structural components. Three interfaces between the system and engineers who use it were described: the SQL query language, database application programs, and the forms menu. Two areas of use were identified: FEM maintenance and FEM-FEA data and results interrogation.

8 TRADEMARKS

NASTRAN is a registered trademark of NASA

MSC/NASTRAN is an enhanced proprietary product of MacNeal-Schwendler Corporation

ORACLE is a registered trademark of Oracle Corporation

CADAM is a registered trademark of CADAM Inc.

VAX is a registered trademark of Digital Equipment Corporation