

**N 8 9 - 2 5 2 2 1**

**AN OVERVIEW OF THE DOUGLAS AIRCRAFT COMPANY  
AEROELASTIC DESIGN OPTIMIZATION PROGRAM (ADOP)**

**ALAN J. DODD  
Douglas Aircraft Company, McDonnell Douglas Corporation  
Long Beach, California**

**PRECEDING PAGE BLANK NOT FILMED**

## INTRODUCTION

This paper describes, from a program manager's viewpoint, the history, scope, and architecture of a major structural design program at Douglas Aircraft Company called ADOP -- Aeroelastic Design Optimization Program. Bruce A. Rommel discusses technical details and current and potential applications of the program in Douglas Paper 8102.\*

The main businesses at Douglas' engineering division are design and analysis of large subsonic transport aircraft structures and development of existing designs. There is also a sustained effort at the advanced design level that involves subsonic and supersonic projects.

Since the mid 1950s, Douglas has maintained a generally uncoordinated research and development effort involving computerized methods for loads, statics, and dynamics. Plans for a comprehensive coordinated computational structural mechanics (CSM) effort were developed by Rommel in 1978. These plans were further refined for the (unsuccessful) proposal effort of 1982 for the Air Force contract which resulted in the Air Force Structural Optimization System (ASTROS). The ASTROS proposal led Douglas management to authorize full-scale development of ADOP, starting in late 1984.

ADOP was originally intended for the rapid, accurate, cost-effective evaluation of relatively small structural models at the advanced design level, resulting in improved proposal competitiveness and avoiding many costly changes later in the design cycle. Before release of the initial version in November 1987, however, the program was expanded to handle very large production-type analyses.

\* Rommel, Bruce A. Meeting the Challenges with the Douglas Aircraft Company Aeroelastic Design Optimization Program (ADOP). Douglas Aircraft Company, Long Beach, California, DP 8102, September 1988. Presented to the Second NASA/Air Force Symposium on Recent Experiences in Multidisciplinary Analysis and Optimization, September 28-30, 1988. NASA CP-3031, 1988. (Page 1369 of this compilation.)

DESIGN AND ANALYSIS OF LARGE TRANSPORT AIRCRAFT  
MODIFICATIONS TO EXISTING DESIGNS  
ADVANCED PROJECTS -- AST, NASP, HSCT, PROPFAN  
1982 -- "ASTROS" PROPOSAL  
1985 -- FULL-SCALE ADOP DEVELOPMENT AUTHORIZED

## REQUIREMENTS

The original Douglas requirement called for the rapid static and aeroelastic analysis and optimization of advanced design layouts. It was assumed that an accurate stiffened-shell, finite-element representation of the layout could be processed as a single entity. The available production analysis programs generally were not automatically interfaced, could not respond rapidly or cheaply enough for proposal or initial design purposes, and did not have optimization capabilities.

As soon as ADOP development started, it became clear that the automatic features of the program -- and particularly the use of a common finite-element model for static and aeroelastic analysis -- could also improve the efficiency and accuracy of production analyses. We were also asked to prepare for operations on a Cray XMP18. Therefore, a large detour was taken in the development schedule to expand the numerical method algorithms and data base utilities for out-of-core use.

It also became clear that our initial decision to use the in-house Computer Graphics Structural Analysis (CGSA) modeler program would limit ADOP applications. On a corporate-wide basis NASTRAN is often used, especially for dynamic analysis, so work was initiated on a NASTRAN-ADOP model translator.

### ORIGINAL REQUIREMENT

- RAPID STATIC AND AEROELASTIC EVALUATION OF ADVANCED DESIGN LAYOUTS

### SUBSEQUENT REQUIREMENTS

- LARGE-SCALE MODAL ANALYSIS
- CRAY APPLICATIONS
- INTERFACES WITH PATRAN AND NASTRAN

## JUSTIFICATION

We were lucky at the start of ADOP planning in that Douglas was expanding rapidly, and our technical managers had research and development backgrounds and were sympathetic to the cause. The figure lists the main points used to justify what is, even for McDonnell Douglas Corporation, a large investment of research funds and manpower.

- Contracts may be won or lost at the proposal level, so it is essential that engineering inputs be as accurate and comprehensive as possible. An important factor at this stage is high-quality graphics.
- The cost of making design changes rises rapidly as the design cycle progresses. A program like ADOP allows correct design decisions to be made very early in the design process so that fewer changes are needed as the design matures.
- One reason that the existing analysis system cannot respond efficiently is the lack of an automated data interface between the loads, statics, and dynamics functions.
- Some managers advocated buying outside code and waiting for ASTROS. In reply, it was argued that to remain competitive it would still be necessary, even if ADOP were not being developed, to maintain a core group of engineer/programmers capable of evaluating and implementing state-of-the-art CSM programs and technology and capable of integrating new code with existing in-house analysis systems.
- None of these arguments would be valid were it not for the fact that significant and inexpensive computing power is now available and matrix methods to take advantage of this power are mature and accessible.

### COMPETITIVE PROPOSAL WORK

A GOOD EARLY DESIGN = TIME AND COST SAVED LATER

NEED AN INTERDISCIPLINARY DATA BASE

NEED TO KEEP ABREAST OF TECHNOLOGY/PROGRAMS

NEED TO KEEP ABREAST OF COMPETITION

TECHNIQUES AND COMPUTING POWER NOW EXIST

## DEVELOPMENT PHILOSOPHY

The objective of ADOP development is to produce user friendly code for the working engineer. There has been much discussion on the issue of user friendliness -- how much should the user be expected or required to know about the code and the methodologies incorporated? Fortunately, the ADOP control language, or ACL "DMAP" approach, allows the user to largely make that decision. He can pick up an existing "black box" ACL sequence or write his own.

Practical considerations dictate that existing modelers be used. We started with the in-house CGSA/CASD modeler but can now process NASTRAN bulk data sets from any compatible modeler.

Similarly, it makes practical sense to interface with existing in-house programs to generate weights and aerodynamics data. Much effort has been expended writing interfaces with these programs and persuading programmers to adopt ADOP data-management utilities for future work.

The main departure from existing practice is in the use of a common finite-element model for static and aeroelastic analysis. Currently, beam models are used for dynamics work, and the translation of data between models is a serious bottleneck. In practice, static analysis models are commonly finer than a one-for-one representation, which is more detailed than necessary for dynamic analysis and very expensive to process. This observation does not yet apply at the advanced design level, but in the future it may be necessary to supplement the static analysis model with a simpler, parallel version for dynamic analysis, or introduce comprehensive global/local analysis features to aid the static analysis of relatively coarse models.

The size of these models dictates that ADOP be optimized for mainframe or supercomputer applications. Douglas is an IBM house that has access to a Cray XMP18. There are no plans to adapt ADOP to mini- or personal computer installations.

As an aid to automating the design process, a comprehensive graphics capability has been specified. This encompasses some pre- and post-processor functions, but the main objective is to provide intermediate processing to aid in data generation, display, and transfer.

Literature reviews and practical experience suggested that the methodology to achieve these objectives was mature and readily available. It was decided to use consultants, when necessary, to ensure that the best available technology was incorporated. In practice, we repeatedly found that development, debugging, and adapting were necessary before existing methods or code could be incorporated in ADOP.

PROGRAM TO BE USER FRIENDLY  
USE EXISTING PRE-PROCESSORS (CGSA, PATRAN)  
INTERFACE WITH EXISTING IN-HOUSE DISCIPLINES  
SAME MODEL FOR STATICS AND DYNAMICS  
PROGRAM FOR MAINFRAMES -- NO PC VERSIONS  
EXTENSIVE USE OF GRAPHICS DISPLAYS  
MINIMAL METHOD DEVELOPMENT -- USE CONSULTANTS

## PROGRAMMING TECHNIQUES

Most of the coding for ADOP is in FORTRAN 77, except for a new, C-coded graphics module called IMAGES -- Interactive Modal Aeroelastic Graphical Engineering System. The coding has been done by engineers rather than programmers, but with an awareness of the rules of structured programming and top-down programming. Most are common sense rules which nevertheless need to be emphasized. In essence, they are-- Plan what you are going to do, do it methodically, test it methodically, and establish programming rules so that the code can be modified and maintained as easily as possible. The system operates on IBM 3090 mainframes, which also act as a front-end processor to a Cray XMP18 at McDonnell Douglas in St. Louis.

Top-down programming concepts were applied to the overall design of the highly modular system, but we found that at the module level, the individual programmer must be free to do it his way -- bottom-up if necessary -- without having to plan every detail of the module before starting to code. The programmer must, of course, use the ADOP disk and memory management system (next page), which dynamically allocates virtual memory for array manipulation and provides for associated disk files. Arrays are tracked and manipulated by a comprehensive set of disk and core management utilities. Arrays are defined in detail (name, dimensions, type, and storage location) and the information stored in COMMON blocks. This is virtually the only use of COMMON in ADOP, and its use is transparent to module programmers.

The modules are called by a high-level ADOP control language (ACL), based originally on the Douglas FORMAT program and similar to NASTRAN DMAP and ASTROS MAPOL. At a still higher level, the ADOP run setup (interactive or batch), source deck updates, compiling and linking, and other functions are controlled by a series of TSO CLIST programs.

MODULAR ARCHITECTURE USING FORTRAN 77

IBM 3090 AND CRAY XMP18

DYNAMIC WORK ARRAY (ALMOST NO COMMON BLOCKS)

WORK ARRAY AND DISK MANAGEMENT UTILITIES

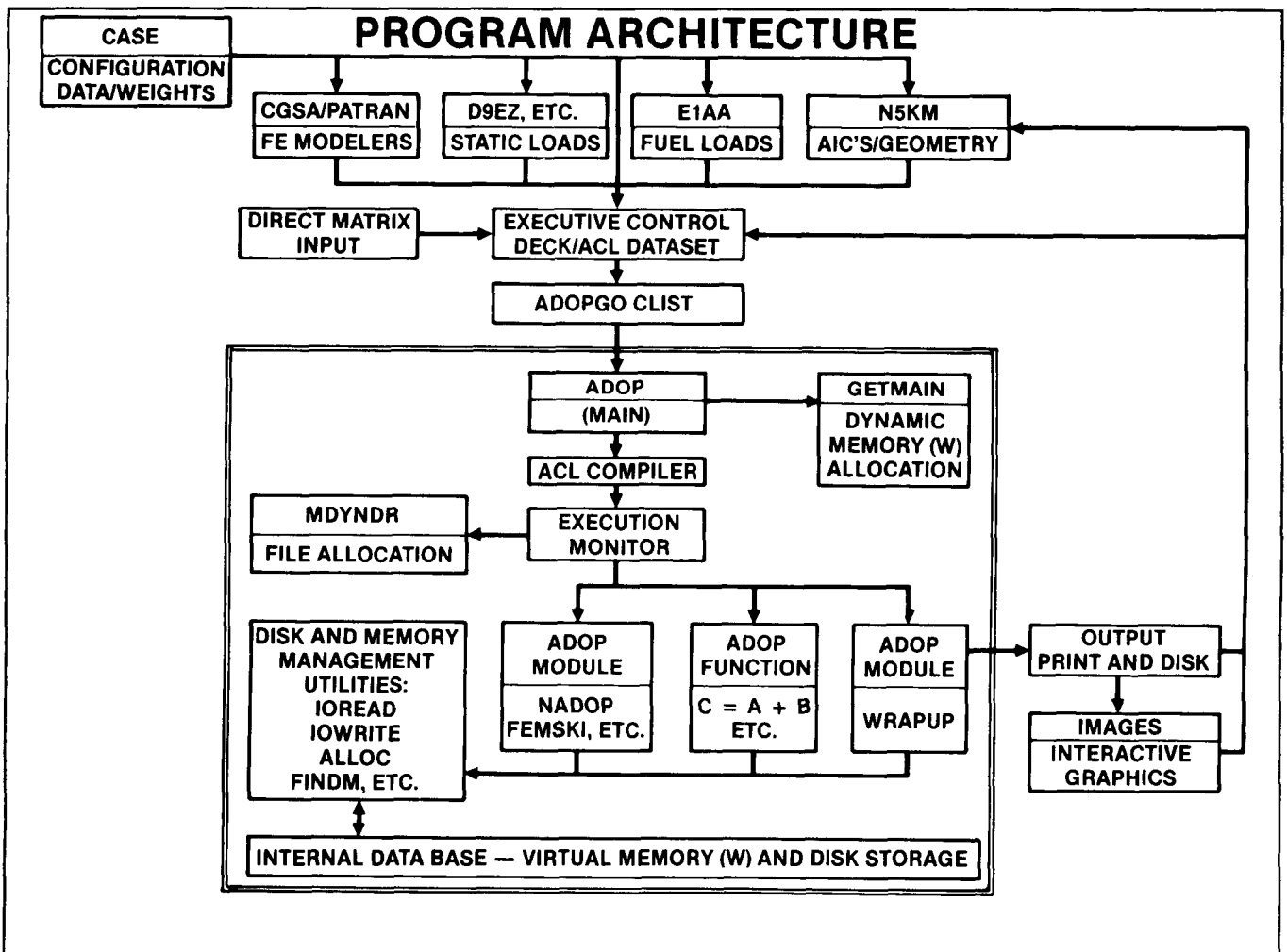
HIGH-LEVEL MATRIX/FUNCTION LANGUAGE (ACL)

PROGRAM CONTROL VIA TSO CLISTS

## PROGRAM ARCHITECTURE

The figure is a simplified depiction of the ADOP architecture, showing the main data handling features and the relationship between ADOP and existing in-house codes. Starting from the top, basic configuration data is fed to the modeler and loads programs. Except for the generation of mass data, via IMAGES, from CASE, this step is not yet automated. Dataset names from the modeler and loads programs are input through ACL instructions and the datasets converted to ADOP format with translator modules. Program N5K was modified to directly produce data in ADOP format. Hopefully, the ADOP utilities will be adopted for new and updated programs. The ADOPGO CLIST asks the user for the ACL instruction dataset and output dataset names, then executes ADOP, either on line or batch.

Within the ADOP block, dynamic memory is allocated, and the ACL instruction list is compiled and passed as executable code to the execution monitor. The execution monitor allocates output and internally used files, and defines input files for the data base. Instructions are executed sequentially, ending with the WRAPUP module, which closes all files, prints lists of arrays on files and in memory, and exits back to the ADOPGO CLIST. Module architecture consists of driver subroutines or "task supervisors," which prepare input to and accept output from the "function subroutines," which ideally should be stand-alone code, independent of the ADOP system. Note that the only way a module should communicate with the data base is through the ADOP utilities.



## GRAPHICS

It was decided early in the ADOP planning process that interactive graphics should be emphasized as an aid to the automation process. After a false start using DYNA-MOVIE-BYU on an E&S 340 work station, we switched to a Silicon Graphics IRIS 3030 work station with a UNIX operating system. The comprehensive Silicon Graphics library firmware was used to write the new code for IMAGES.

The work station has sufficient capacity to operate independently of the IBM mainframe (unlike the E&S 340), but is linked to the IBM for file transfers to and from ADOP. A Seiko copier produces high-quality color screen images on paper or transparencies.

IMAGES performs the following functions:

- Display of the assembled FE model to allow visual checking for modeling errors and inconsistencies
- Definition of mass and weights data. Weights data is translated from the CASE model to the ADOP structural model and redefined as distributed or concentrated masses. Masses from other sources can be added at this stage. This data may be developed interactively on the IBM system, but the process is made much easier with the aid of IMAGES.
- Animated display of mode shapes and static deformations
- Color stress contours
- Graphical splining
- Aerodynamic data points generation

Some of these features -- model display and animated modes -- were initially regarded, perhaps cynically, as management and PR display tools, but they have become useful and popular for verifying the model and analyzing results. The animated modes display in particular has proved very useful for spotting "junk" modes, isolating modeling faults, and interpreting "good" modes.

SG 3030 IRIS WORK STATION (FIRMWARE-BASED, C LANGUAGE)

MODEL DEBUGGING

WEIGHTS/MASS DEFINITION

ANIMATED DISPLAY OF MODE SHAPES

GRAPHICAL SPLINING

AERODYNAMIC MODEL GENERATION



## NUMERICAL METHODS

It is not the purpose of this presentation to detail the numerical methods incorporated in ADOP. The figure lists the chief methods currently operational. Future plans include comparisons of methods for accuracy, efficiency and applicability -- for example, Lanczos versus subspace iteration for modal analysis; optimality criteria versus numerical search methods for structural optimization.

As noted previously, it is often difficult to judge the relative merits of different methods just by studying the literature. Generally, the differences between established methods are not great and competing techniques often can be shown to differ only in the numerical "tricks" that are played.

The main area for improvement lies in the strategies employed in the use of the numerical methods -- for example, the wavefront method for solving the finite-element model equations versus the blocked skyline approach; the modal reduction approach for aeroelastic loads calculations versus calculations in discrete structural coordinates.

### DISPLACEMENT METHOD STATIC ANALYSIS

- WAVEFRONT EQUATION SOLVER
- FULLY STRESSED DESIGN
- CONSISTENT OR DIAGONAL MASS MATRIX

### LARGE ORDER MODAL ANALYSIS

- ACCELERATED SUBSPACE ITERATION
- RESTART
- INDEPENDENT RIGID BODY MODES
- NUMERICAL ERROR MONITORING

### DYNAMIC RESPONSE

- DIRECT OR MODAL TRANSIENT RESPONSE
- AUTOMATED TIME HISTORY LOADING

### GRAPHICAL SPLINING - HARDER, BEAM SPLINE

### K-METHOD FLUTTER LOOP

- DAMPING VS. REDUCED FREQUENCY

### NUMERICAL SEARCH

- ADS, CONMIN, ETC.
- SENSITIVITIES

## SUMMARY

An Aeroelastic Design Optimization Program (ADOP) is currently being tested at Douglas Aircraft Company.

ADOP was originally intended as a relatively small-scale program to improve structural evaluations at the advanced design level. Soon after full-scale development began in 1985, it became clear that a large-scale capability was required and ADOP was expanded so that problem size is now limited only by available machine resources or cost.

ADOP drew upon experience with FORMAT, NASTRAN, and various in-house dynamics and flutter codes. The aim has been to incorporate state-of-the-art program architecture, numerical analysis, and nonlinear programming methods. Currently the main features of ADOP are

- A comprehensive disk and memory management system
- A high-level function language, ACL
- Large-order finite-element and modal analysis modules, FSD resizing and an automated flutter loop
- A graphics module, IMAGES, for intermediate data processing

To be accepted, a new program must make the analysts' job easier or have capabilities not otherwise available. ADOP offers these features, is well documented, reasonably user-friendly, and backed by expert assistance. In addition, the user may use modelers with which he is familiar.

Development is continuing...

ADOP IS UNDER TEST AT DOUGLAS AIRCRAFT COMPANY, LONG BEACH

EXPANDED FROM SMALL ADVANCED DESIGN CODE TO LARGE-SCALE GENERAL  
PURPOSE SYSTEM

STATE-OF-THE-ART AEROELASTIC ANALYSIS/OPTIMIZATION FEATURES

USER-ORIENTED

DEVELOPMENT IS CONTINUING