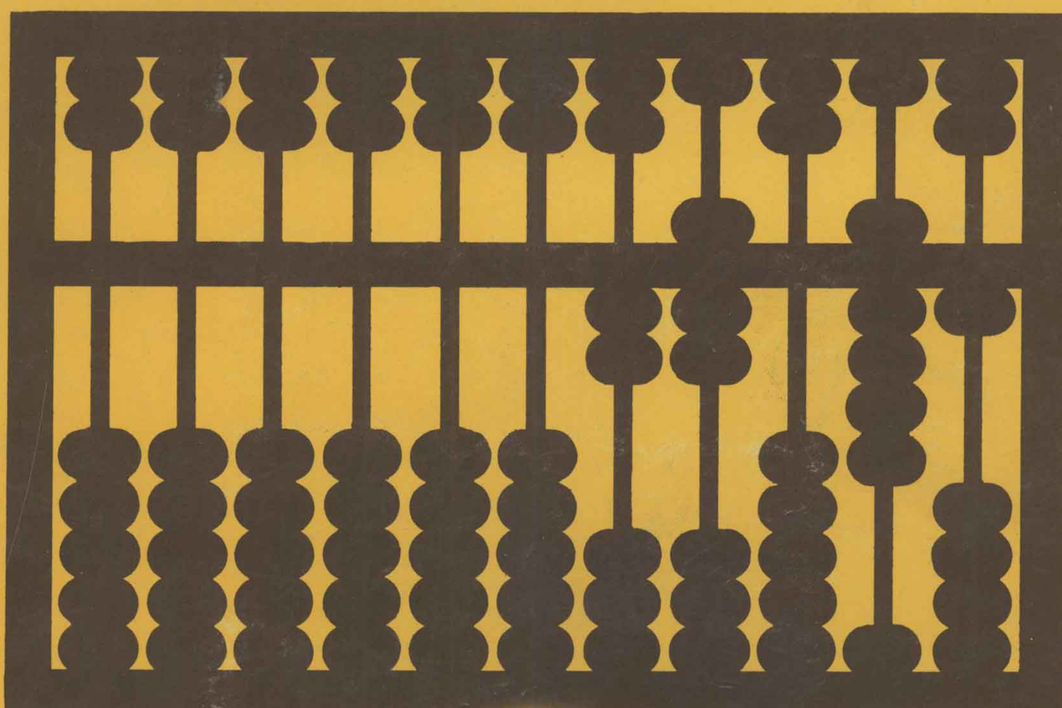
# INTERNATIONAL CONFERENCE ON EDUCATION, PRACTICE AND PROMOTION OF COMPUTATIONAL METHODS IN ENGINEERING USING SMALL COMPUTERS (EPMESC)

SPONSORED BY THE INTERNATIONAL ASSOCIATION OF COMPUTATIONAL MECHANICS

Macau, 5-9 August 1985



**SPONSORS**

The Government of Macau
The University of East Asia, Macau
The Technical University of Lisbon
The Institute of Industrial Science, University of Tokyo
The Asian Institute of Technology, Bangkok
The National University of Singapore
The United Nations University
UNESCO (R. O. For S&T – Southeast Asia)
The International Association for Computational Mechanics

**ORGANISING COMMITTEE**

E. R. de Arantes e Oliveira
Y. K. Cheung
P. Karasudhi
T. Kawai
S. L. Lee
Luo Songfa
T. H. Pian
S. Valliapan

VOLUME 1A
(August, 5th, A. M.)

# VOLUME I

(AUG. 5th)

# COMPUTER-AIDED DESIGN AND GRAPHICS

# IMPLEMENTATION AND TRAINING REQUIREMENTS OF A MODERN INTERACTIVE ENGINEERING PROGRAM ON SMALL COMPUTERS

Henry H. Fong

PDA Engineering
1560 Brookhollow Drive
Santa Ana, CA 92705-5475, U.S.A.

## ABSTRACT

PDA/PATRAN® is a state-of-the-art, interactive engineering program with solid modeling, finite element modeling, and stress analysis capabilities. Based upon an analytic solid modeling (ASM) mathematical foundation and an integrated data base, it operates on the most advanced computer and color graphics display hardware available today. The recent successful penetration of the vast 32-bit supermini market (as exemplified by industry standard Digital Equipment Corporation's VAX®) by the supermicros (typified by Apollo) has caused keen concern among the supermini manufacturers. This paper highlights some of the experiences of one software vendor (PDA Engineering) in implementing its PATRAN program on the current generation of small computers and graphics devices, and the training and support activities required to meet a growing demand.

## THE PATRAN SYSTEM

PATRAN consists of solid modeling, finite element pre- and post-processing, and (optional) stress analysis capabilities—all using the same data base. Figure 1 illustrates a combined view of a solid shaded model of a piston body together with the finite element model of the connecting rod and sculptured head. PATRAN uses ASM, a solid modeling technique which is different than the "constructive solid geometry" and "boundary representation" methods [1]. Lines, surfaces, and solids are described as continuous *parametric cubic* functions. The generality, speed, and accuracy of the ASM mathematical framework lend PATRAN to mass property calculations, finite element meshing, imaging, animation, and the design of composite materials.

Once the geometry model has been generated, PATRAN offers the engineer a large number of highly automated options for creating a finite element model. PATRAN provides uniform and non-uniform mesh generation and transitioning capabilities (Figure 2). It supports the most commonly used linear, quadratic, and cubic finite elements. The material property synthesis capabilities are extensive and sophisticated, and include isotropic, orthotropic, and anisotropic materials, as well as laminated and solid composites. Loads and boundary conditions can be concentrated or distributed. PATRAN provides instant 3D graphic feedback to the engineer to verify their proper usage (Figure 3). Physical properties (such as thickness) can also be constant or varying. To verify the suitability of element shapes in the finite element model, the engineer can access model verification options to check element aspect ratio, skew, taper, and warp. The engineer can also check for free edges and faces, cracks, or reversed surface normals. Before submitting the model for analysis, the engineer can ask PATRAN to minimize the model bandwidth or wavefront in order to reduce analysis costs.

PATRAN offers an optional STRESS analysis module for linear static and vibration analyses. The structural integrity of a design can be simply assessed by a designer/analyst, without ever leaving the PATRAN system and without the need for translators linked to other analysis codes. Modifications to the design can be easily made and evaluated. This interactivity turns the computer into a computational laboratory environment, where the engineer can experiment and gain insight. The user can combine high-priority interactive modeling with analysis execution in the background. Up to a maximum of five jobs can be controlled simultaneously.

Post-processing of analysis results is dramatically enhanced by the use of color. Deformed geometry can be plotted with or without hidden lines. Element results (strain, stress, etc.) can be color-coded and plotted, as

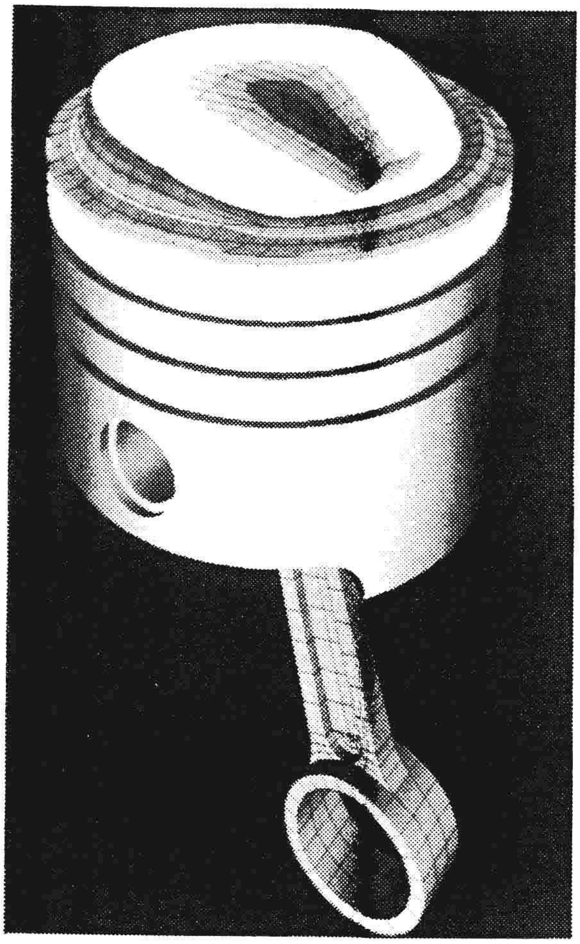


*Figure 1. Combined view of solid model of piston body, finite element model of connecting rod, and color fringe plot of applied temperature loads on the sculptured head in a piston and connecting rod assembly (PDA/PATRAN).*

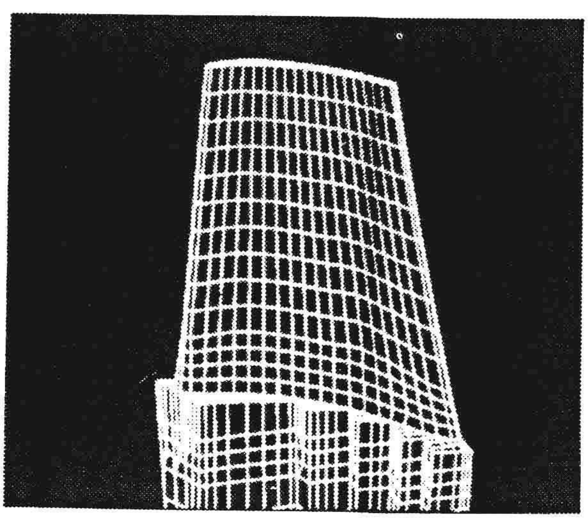


*Figure 2. Finite element model of a single turbine blade showing solid mesh transitioning.*

can their contours. Mode shapes or deformations can be animated; a new capability is turning these animated images into a video movie to enhance realism and impact. A carpet plot may be created from data surfaces. In addition to contour plots, color fringe plots (solid-color contours) are available. Another vivid display is to plot failure criterion (e.g., Tsai-Wu failure criterion for a composite material) on top of a model whose environmental loads (e.g., temperature fringe plot) are shown. And, of course, the superposition of subcase results is possible.

PATRAN communicates to and from most analysis codes and CAD systems using a flexible neutral system. Currently, 26 translators exist to interface with finite element, finite difference, and boundary element codes used for structural and thermal analyses [2]. There are also eight CAD interfaces, including the de facto U.S. geometry standard called the Initial Graphics Exchange Specification (IGES) format.

## IMPLEMENTATION ON SMALL COMPUTERS AND GRAPHICS DEVICES

The successful development of a high-quality interactive graphics program such as PATRAN would not have been possible without two concurrent hardware breakthroughs in the late 1970s: the 1978 debut of Digital Equipment Corporation's (DEC) VAX 11/780 32-bit, virtual memory supermini (and the vast industry it spawned), and the subsequent proliferation of relatively inexpensive high-resolution color display devices. While two to three years ago the industry standard among raster and vector graphics devices was 16 colors and a resolution of 512 x 512 pixels, today the industry is moving toward 256 simultaneous shades of color and 1024 x 1024 resolution, with some devices offering a palette of 16.7 million colors and 1500 x 2000 resolution. Indeed, many engineers these days have added the following graphics jargon to their vocabulary: solid shading; anti-aliasing; jaggies; flicker; hue, light, and saturation (HLS); multiple light sources; color lookup tables; direct memory access (DMA); Phong and Gouraud shading, etc.

In July 1978, PDA was chosen by DEC to have the first VAX 11/780 installation on the U.S. West Coast [3]. The interactive nature, multiple user capability, speed, virtual memory, and other user-friendly features of "the VAX" quickly established it as the minicomputer industry standard, a position which it still enjoys. This leadership position

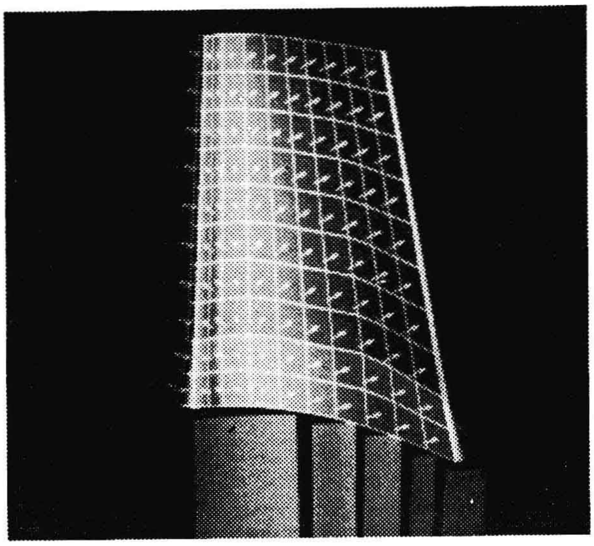


*Figure 3. Three-dimensional graphic feedback of applied pressure loads on blade model.*

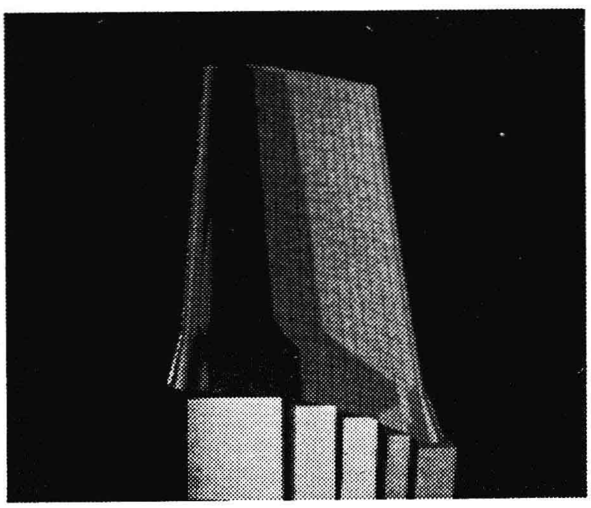


*Figure 4. Evaluating analysis results in a color-coded image superimposed on the blade itself, to highlight the structural response to the operational loads.*

is borne out by two recent observations: (1) numerous claims, performance benchmarks, and cost comparisons cited in advertisements by competing computer manufacturers—plus, interestingly enough, self-directed claims by DEC that its VAX 11/785 is 50 to 70 percent faster while the new 8600 is 4.2 times faster than the you-know-what; (2) the author's February 1985 survey of 23 finite element code developers—16 commercial codes, 7 academic or R&D—showed that when asked the question, "which is your development machine", the responses revealed that the clear winner was VAX with 11 votes, followed by IBM (five votes) and a three-way tie among CDC, Prime, and Harris.

Meanwhile, graphics industry leader Tektronix (which was comfortable with its ubiquitous green-on-green 4014 storage tube device in the 1970s) began to lose market share to new color challengers like Ramtek, Megatek, Lexidata, and so forth. Tektronix did not stand still for long, however, and has made the very successful 4115B the industry standard. Now, the market is even more crowded, as new and old vendors such as Evans and Sutherland, Adage, Sigma/Lundy, Raster Technologies, Seiko, and Silicon Graphics go after more local intelligence and the elusive 3D world. It's a truism that the graphics industry is changing even more rapidly than the volatile computer industry.

In 1982, the supermicro and engineering workstation industry was born with the enormously successful Apollo DOMAIN® computers. These processors were smaller, less expensive, networked, and offered 32-bit precision and performance comparable to the superminis. Many companies, large and small, decided to abandon mainframes and superminis and opted for the supermicros instead. Others in this expanding market include new startups like Sun Microsystems, Cadmus, Zilog, Logisticon, Altos, Ridge, Celerity, as well as familiar names such as Hewlett-Packard (HP 9000) and Data General (DG 10 and 10/SP). Their successes made the market more competitive and spurred the supermini vendors to develop and market new, improved, less expensive and smaller versions of their machines. These include, for example: DEC (VAX 11/750, 11/730, 8600, MicroVAX); Prime (9955, 9750, 9650); Data General (Eclipse MV/10000, MV/8000, DS/4200); IBM (4341, 4381-1, etc.); Harris (H-800, H-60); etc.

Table 1 summarizes the implementation history of PATRAN on various computers and graphics devices [3]. Major PATRAN releases occur about once a year. Each time, many new capabilities are added (often at suggestions of users), with more conversions on different computer systems and graphics devices. Currently, PATRAN operates on more than 55 graphics devices and 10 computer systems. In order to help decide among the different computers, PDA's *evaluation criteria* include such items as:

*The Company*

- — management caliber
- — industry position and image
- — financial strength
- — potential market
- — innovation
- — technical support quality
- — marketing expertise
- — company philosophy
- — niche in marketplace
- — long-term cooperation benefits
- — key people (and turnover rate)
- — ambience

**TABLE 1. IMPLEMENTATION OF PATRAN ON VARIOUS COMPUTERS AND GRAPHICS DEVICES**

| Date | PATRAN Version No. | Computers | Graphics Devices |
|---|---|---|---|
| November 1978-August 1981 | 1.0 to 1.2G | DEC VAX 11/780 | Ramtek 6200A<br>Tektronix 4014<br>DEC VS-11 |
| August 1981 | 1.3 | DEC VAX 11/750<br>Prime | Megatek 7200, 7250<br>Ramtek 6211<br>Evans & Sutherland MPS-2 |
| August 1982 | 1.4 | Apollo | Ramtek 6212, 9400<br>Tektronix 4113<br>MVI7<br>Evans & Sutherland PS-300 |
| September 1983 | 1.5 | Control Data Corp. | Lexidata 2410, Solidview<br>Adage<br>Sigma/Lundy<br>VT125<br>Raster Technologies Model 1 |
| July 1984 | 1.6 | Data General<br>Harris<br>Cray<br>Hewlett-Packard HP9000 | Extended families<br>Tektronix 4100 family<br>Raster Technologies<br>Envision<br>Seiko<br>Evans & Sutherland PS-340<br>Silicon Graphics IRIS |
| May 1985 | 1.6C | Tektronix 6200 Series Workstations<br>IBM 4300 Series | Ramtek 2020<br>Adage 3000<br>IBM 5080 |

*The Computer*

- — computer CPU
- — amount of cache (KB)
- — performance
- — clock rate (MHz)
- — operating system design
- — pipelining
- — bus width (bits, rate)
- — word size (≧ 32 bits)
- — virtual memory addressing (gigabytes)
- — maximum physical memory (megabytes)
- — color monitor availability and quality (DMA, max. RS-232 speed)
- — uniqueness
- — packaging
- — networking
- — maximum number of users
- — typical pricing, price/performance

A similar list exists for graphics devices.

## TRAINING AND SUPPORT

Once a sale has been made, to ensure maximum customer satisfaction and a high license renewal rate, the customer support must be top-notch. At PDA, the Customer Services Department performs these functions: hotline support, education and training, documentation, quality assurance, and distribution. These activities are now briefly reviewed.

The PATRAN hotline is supported from 7 a.m. to 5 p.m. (California time), Monday through Friday. This activity has consistently received rave reviews by satisfied customers, who like the fact that over 80 percent of the calls are answered on-the-spot. Recent statistics indicate a rapid increase in calls in keeping with the growth of the customer base. In 1984, the average number of calls per day has steadily increased from 13 in January, to 23 in July, and to 28 in December. In April 1985 the average number of calls is about 50 to 60 a day.

A new East Coast sales and support office opened in the Washington, D.C. area in September 1984. In the summer of 1985, PDA plans to open a European office in London, to be followed in a few months by Munich. In Asia, PDA has entered into a joint marketing agreement with DEC covering Japan, South Korea, Hong Kong, and Singapore, and currently is exploring or entering into agreements with other U.S. computer manufacturers and Apollo distributors in Taiwan and Hong Kong.

Training activities at PDA have grown tremendously to keep pace with the demand. In 1980, the three-day PATRAN introductory seminar (half lecture, half lab) was offered three times a year. This rate increased to bimonthly in 1983, and monthly in 1984, with occasional seminars given on the East Coast and in-plant. Figure 5 shows the number of people trained by the end of 1984 and projected for 1985.

In addition to the PATRAN introductory seminar, current PDA course offerings also include the PATRAN advanced seminar and the STRESS analysis module seminar. This summer, the slate will be expanded to include new seminars on: composites design/analysis, finite element modeling, CAE technology for management, and MSC/NASTRAN usage. Training is an important issue, for it leads to improved productivity and correct use of the software.

The documentation quality of most finite element software packages is usually mediocre to poor. For an interactive engineering system like PATRAN, developing quality documentation is even more difficult and challenging. Current PATRAN "documentation" consists of four types: a 2-volume reference manual, a primer (introductory seminar notes), a pocket command guide (...immensely popular), and a three-level on-line HELP system. In the future, a series of primers are planned on special topics (e.g., geometry modeling, analysis modeling, post-processing), which are portable, readable, and well-illustrated. And, of course, interface documents must be available for all interfaces to analysis codes and CAD systems.

Quality assurance and distribution are also important. The author's February 1985 survey showed that most of the 23 finite element code developers surveyed rated themselves "not automated at all" or "a little bit automated" in their quality assurance procedures; PATRAN is no exception. An interactive system like PATRAN gives rise to an extraordinarily complex set of functions and combinations of computers and

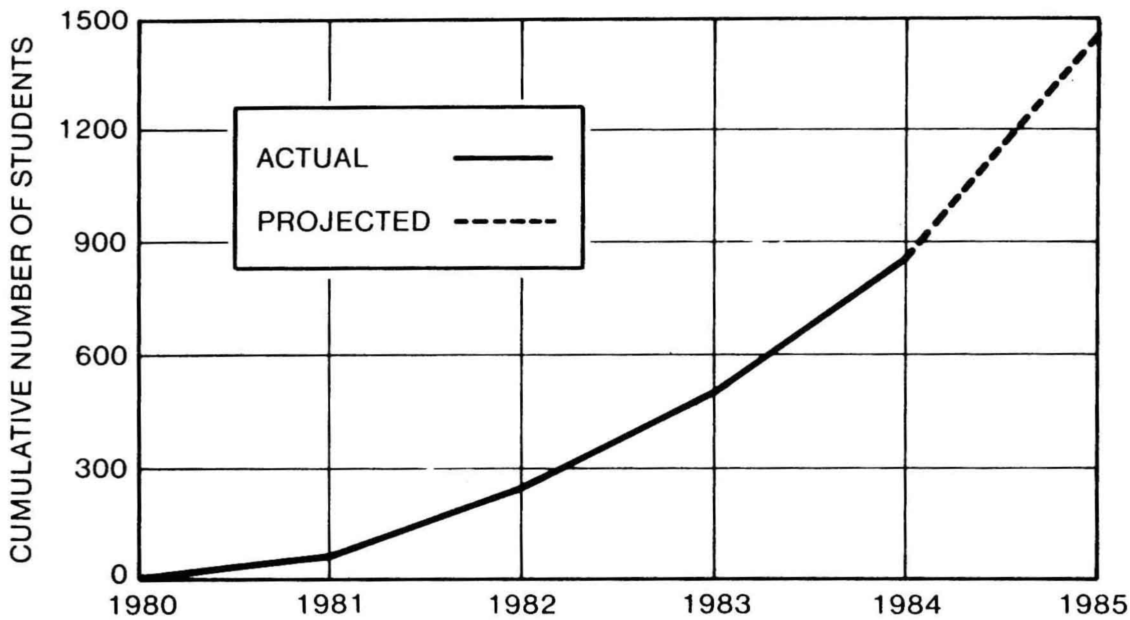



*Figure 5. PATRAN training by calendar year.*

graphics devices which must be checked. Good distribution means sending a customer the right tapes or floppies for his computer and graphics device—fast. Both quality assurance and distribution procedures lend themselves well to automation, and improvements along these lines are under development.

## CONCLUDING REMARKS

The careful implementation of engineering software on different hardware configurations and thoughtful training and support activities are of crucial importance in today's and tomorrow's marketplace. *Portability* is the key to survival, with the arrival of new hardware advances and computer architectures. The next decade will witness significant developments in computer-based instruction (using video discs), artificial intelligence, and expert systems. Quality engineering software, needless to say, must take full advantage of all of these.

## REFERENCES

1. Casale, M. S. and Stanton, E. L., "An Overview of Analytic Solid Modeling", *IEEE Computer Graphics and Applications*, February 1985, pp. 45-56.

2. Fong, H. H., "Interactive Graphics and Commercial Finite Element Codes", *Mechanical Engineering*, June 1984, pp. 18-25.

3. Crain, L. M., "PATRAN, A Historical Perspective", *Proceedings–2nd PATRAN Users Conference* (ed. H. H. Fong), Newport Beach, California, May 21-22, 1984, sponsored by PDA Engineering, Santa Ana, California.

# THE MICRO-COMPUTER AS A DESIGN TOOL

Dr. ir. E. BACKX

Catholic University of Louvain, Belgium

ABSTRACT

Although popular software such as multiplan, autocad, dbase II can be purchased at a fraction of the development cost due to the large amount of interested users, it is more difficult to find highly technical software at comparable prices. The result is that the software for engineering applications can be as large or many times larger than the cost of the computer. The paper describes how the software development coŝt can be reduced by using a scientific database manager. This database mahager has been written in basic, fortran 77 and Pascal and has been used in finite element programs, boundary element programs and a steel design package on mini and micro computers. The database manager becomes the core of all programs starting from preprocessors up to the programs that generate finished drawings and tables in the design process.

## 1. INTRODUCTION

Writing large programs for scientific computations such as finite element and boundary element programs and pre- and post-processors is a time demanding job. Every effort has to be made to decrease the number of dependencies of one module to another module because these links are often the reason of bugs. Storage allocation both incore and out of core is one of these problem areas that oblige the programmer to be inventive and very disciplined. Dynamic incore storage allocation in FORTRAN programs is sometimes solved by using a one dimensional array in COMMON. During execution, data arrays are given the required space in this array and the calls to subroutines is done by passing the appropriate pointers. The space used by arrays which are not needed anymore, can be reused by other arrays if the size is smaller than the previous array. To avoid errors in reusing the space of deleted arrays, the data needed have to be well structured before programming is started. Additions to the programs become tedious.

When the programs data needs become too large to justify a complete incore solution, the data have to be transferred to disk files for intermediate or permanent storage. Some programs need as much as 20 different external files to solve this problem.
This is a very unsatisfactory solution for the programmer (and the user).

A remedy for these problems is to use a data manager. Data base managers are well known in administrative applications but not so much used in scientific applications. The reasons are that these data base managers are very heavy programs in their own and due to their capabilities, they tend to decrease drastically the execution speed of the application. Moreover, these data base managers are expensive.

This paper proposes the use of a scientific data manager to solve the problems mentioned above. This scientific data manager is by no means comparable in size and capabilities to a commercial data base manager but it is shown from experience that it is a valuable tool in program development.

## 2. DESCRIPTION OF THE DATA MANAGER

The intention is to free the programmer from managing memory and disk space organisation by giving him easy to use commands for storing and retrieving data. Both incore and out of core data management is taken care off. From the application programmers view, the use is very simple. There are only five different functions he needs to know :

1. Open the database
2. Close the databse
3. Put a data item in the database
4. Get a data item from the database
5. Delete an array in the database.

The way to execute one of these functions is different in the different implementations. At this moment, five different versions have been written:

1. BASIC-2 on Wang
2. MS Basic
3. Fortran 77 on Vax
4. Pascal on Vax
5. MS PASCAL

The data manager controls the data organisation in core and out of core (diskspace). When a particular data item is needed, a request is sent to the data manager and the value of the data item and/or the address in core is returned. Similarly, a data item can be added to the database. The data manager optimises the use of memory and the accesses to the disk. Data items are characterised by a name and a sequence number.
The manager is based on a paging system. When sufficient memory is available to hold all pages containing data, there is no need for disk input - output and the application program will experience only a minor decrease in execution speed when compared with a conventional data structure using arrays.

## 3. IMPACT ON PROGRAM DEVELOPMENT

The use of a data manager in new developments has following advantages :

- reduced development time
- no restrictions imposed by memory size
- maintenance and modifications are much easier.

This can be illustrated as follows. Suppose that you have a program for space structures. The result of this program is a list of displacements and beam forces for a set of load cases. You would like to add a small program to check if the resulting force from a combination of load cases is smaller than the allowable buckling force. In that case, you will need the length of the beam, the profile properties, the material properties, the forces in that beam for all load cases which contribute to the combination. In a classical approach, the new program will start by loading all data needed from files which have been created by the space structure program.

This means that you have to know how these data have been written to disk, which files, what format. Moreover, all data has to be loaded, even if only a few items will be used. This can take time and requires some coding and checking. When the space structure program uses a data manager however the programmer simply sends a request to the data manager to get the required data items and he can proceed immediately with the program.

4. SIZE LIMITATIONS.

Most of the problems caused by memory size are solved by using the data manager. However, there are still limitations. The programmer has to choose a page size, the memory size that can be used for data as a number of pages and the disk size that can be used for data as a number of pages. The size of the page depends on the data to be managed. If there are only a few large arrays, the page size can be relatively large. However, if there are a large number of smaller arrays, it is better to choose a page size equal to the size of the smaller arrays because a large page size will lead to an uneconomic use of memory and disk space. It is always advantageous to choose a large number of pages for the incore data. This reduces the number of disk accesses and therefore increases execution speed.
The minimum number is however 2.

Exemple : page size = 5120 bytes
nr. pages incore = 10
nr. pages on disk = 1000

The total amount of data that can be managed is 5MB.
The total amount of memory required is 57KB.

Special routines have been written to allow transportability of the data. This means that a data base can be started on a very small computer using a floppy and afterwards, the same data can be used on a larger computer using a hard disk. These routines allow a change in number of pages incore and on disk. However, a change in page size is not allowed.

5. APPLICATIONS.

The BASIC-2 version is implemented in the programs of SCIA*.
These are finite element and steel structure design programs.
The FORTRAN version is used in our university in a boundary element program and is also used to couple FEMGEN (a finite element preprocessor) with other programs. The finite element mesh generated by FEMGEN is stored in our database. The input for the finite element program is generated from these data, the bandwith is minimised, the structure is computed and the results are stored in the same database. The post-processor program only needs this database as an input to present the results in a most convenient way.

* SCIA S.V., Steenweg 108, 3912 Herk-de-Stad, Belgium

## 6. RELATION WITH DESIGN.

In design, different phases can be recognised, starting from a preliminary design, than the detailed design and finally, the planning. When a micro-computer is to be used as a design tool, than it must be able to cover all phases of a design project. Several different programs will be used to make analyses, to produce drawings, to do economic calculations, to manage the project.
It is essential that data should not be scattered around so that each time data has to be reentered and stored. This leads to redundant data and is a source of errors. The proposed data manager can help to integrate all these programs around a unique project database. It is, however, not suited to act as a central database for a company. The structure is too simple and only a few querying functions are supported to keep searching time down and code length small.

# INTEGRATED SOFTWARE SYSTEM FOR STRUCTURAL ANALYSIS AND DESIGN

Worsak Kanok-Nukulchai+

Asian Institute of Technology, P.O. Box 2754, Bangkok 10501, Thailand

## INTRODUCTION

Analysis and design of structures constitute the two major activities in structural engineering practices. Generally, these are two mutually coupled processes. While design variables affect the analysis, results of the analysis dictate the choice of these variables in the design. Traditionally, analysis and design are handled as though they are two independent processes. In theory, a trial and error procedure is necessary in order to satisfy the requirements of both at the same time. The attempt to adhere to this consistency by hand calculations, however, may require such a tremendous effort. It is not worthwhile nonetheless if member designs are merely based on the result of a simplified analysis. The provision of a factor of safety is assumed to help offset errors caused by stress misrepresentations as a result of analysis over-simplifications. This eventually leads to a design in which the margin of safety fluctuates greatly from place to place. Such a design is often far from optimality.

After the advent of computers, a rigorous analysis can be automated with great generality using the finite element method (FEM). This is possible because of the objectivity of the structural analysis process. An exact algorithm can always be established without ambiguity.

For design, computer-automation is not feasible due to the lack of a unique algorithm for design synthesis, as much as the lack of 'engineering sense' in computer. Lacking intelligence, computer has great durability, precision and a perfect memory. Computer-aided design (CAD), therefore, aims at combining the roles of human designer and computer, using the strengths of both to complement and supplement each other.

The objective of an integrated software system presented in this paper is not an attempt to couple structural analysis and design into one process, but to 'streamline' their interaction through the use of a standardized data structure. Their traditional distinction as well as their logical orders remains unchanged. A data bank is employed to pool all data units at one place, accessible and amendable at various phrases of analysis and design operations.

The idea of an integrated FEM/CAD system is not new. Over a decade ago, FEM/CAD was the exclusive province of large aerospace companies. The long occupation of a time-sharing system during a CAD session may be too costly for smaller engineering firms. Now this situation has changed drastically with microcomputers. Despite the low microprocessing speed, the insignificant operating cost as well as the attractive input/output capabilities often makes FEM/CAD on microcomputer very effective. In practice, the low processing speed does not necessarily pose a problem as human responses during a CAD take much longer time, whereas lengthy FEM operations can always be unattended overnight. It is believed that future

+Associate Professor of Structural Engineering professional activities of most structural engineers will center around an FEM/CAD work station, similar to the artist's impression in Fig. 1. Figure 2 further illustrates how basic activities of a small consulting firm can be computerized in the modern age.

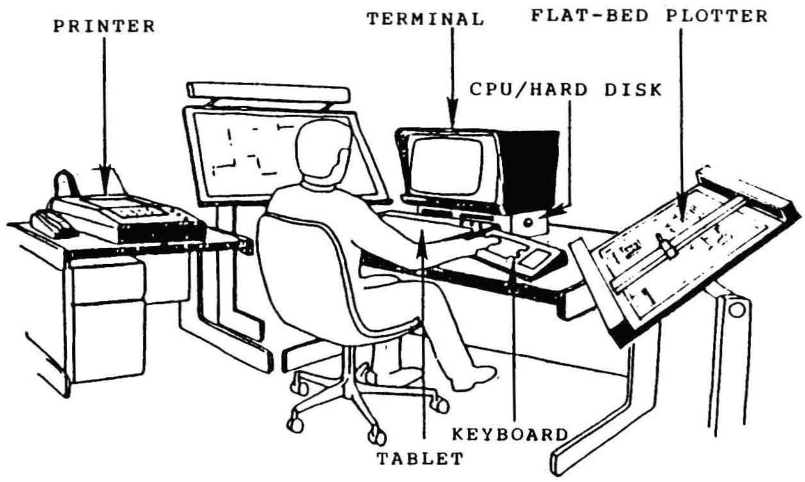



**Fig. 1. Artist's impression of structural engineer's work site**

Today, professionalism in microcomputer software writing is more critical than the mainframe counterpart. Optimizations of program and data structure in relation to a hardware environment are absolutely crucial, to enhance the solution capacity as well as alleviate excessive demands for microcomputer time and human attendance. This paper attempts to outline some useful software development strategies, believed to be necessary for a practical FEM/CAD microcomputer program.

## STRUCTURAL MODEL

Rigorousness of structural modeling depends largely on the computing tool available to the structural engineer. With manual calculations, one can either simplify the structure into a model for which a solution is available, or retain its rigorous form but strives to obtain an approximate solution. In both cases, correctness of the result is often questionable. With computer, it is possible to formulate a more representative structural model, which will reflect behaviors of the real structure more closely and comprehensively.

In the past 25 years, the evolution of FEM has been so phenominal that today, FEM is already considered as an industrial standard for structural analysis. A new challenge to structural engineers in the modern age is the ability to implement FEM and solve large-scale practical problems on office microcomputer.

## ANALYSIS AND DESIGN INTERACTION

Consider a structural model as an assemblage of structural elements (members) interconnected at nodes (joints). An element may belong to any of the standard element types, such as beam, column, etc. Several elements of the same type can have or be assigned similar values of material properties and design variables, and thus together constitute a member design group.

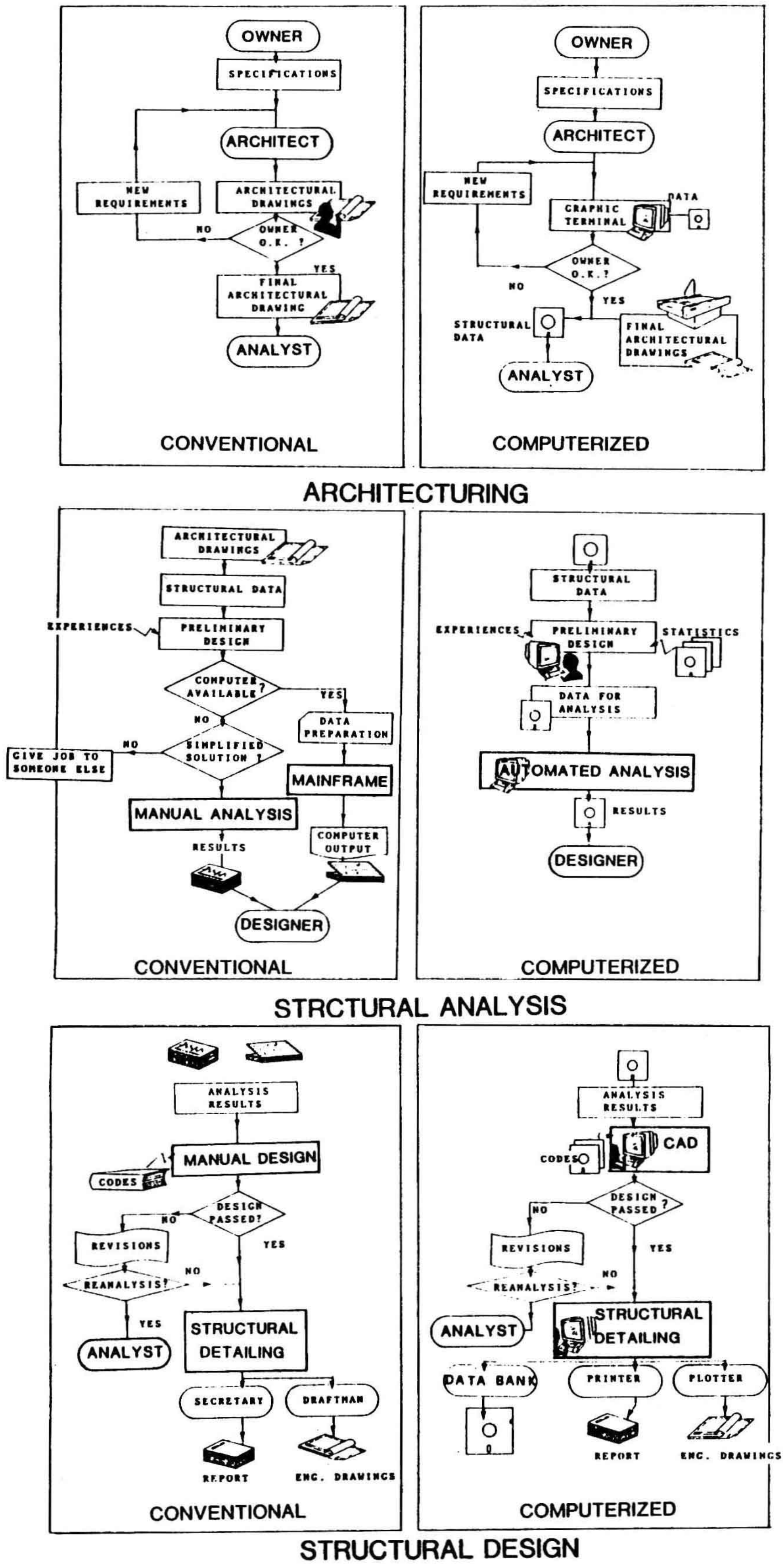



Fig. 2. Computerization for structural engineering activities.

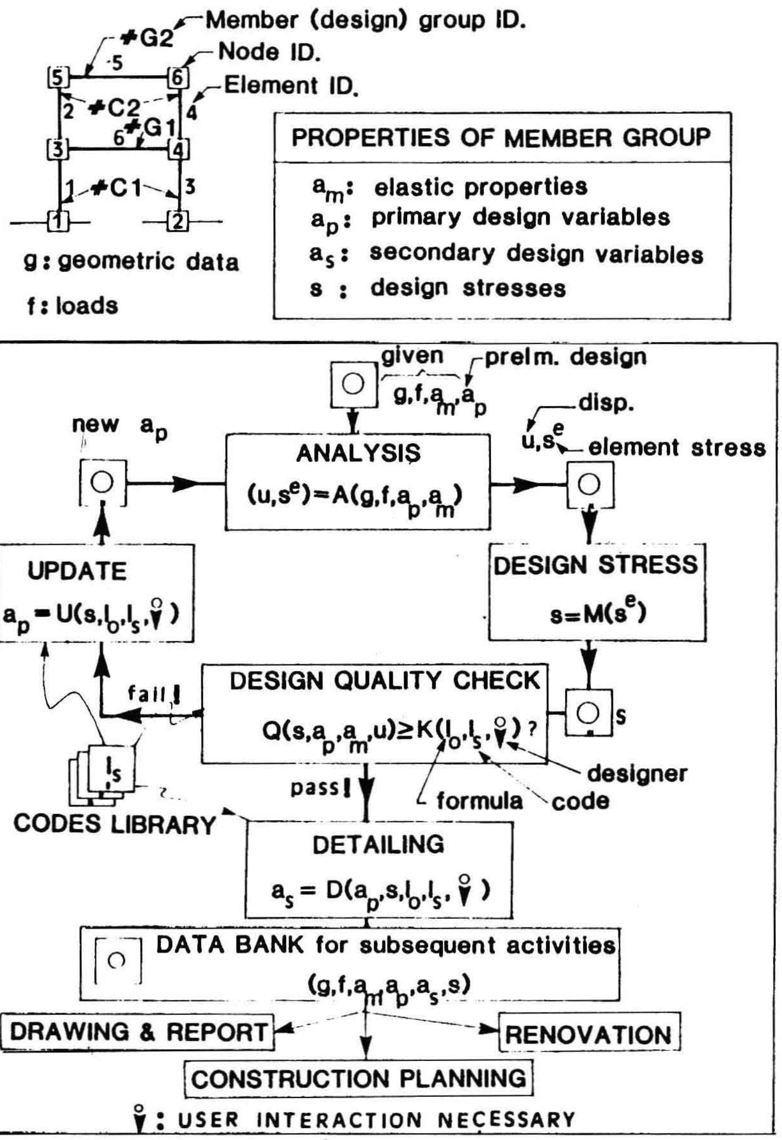



**Fig. 3. Interactions among activities of structural analysis and design**

Typical activities for completing the entire process of analysis and design of a structure, as well as their data interactions are presented in the diagram of Fig. 3. With reference to this diagram, the following points are emphasized:

1. Primary design variables are quantities which affect analysis as well as design strength capacities, e.g. dimensions of cross-section, bar areas etc. The secondary design variables such as bar sizes, numbers and their placements, are determined afterward from the primary design variables, simply for the purpose of detailings.

2. The acceptability of a design depends basically on two types of supplementary information ($I_o$ and $I_s$) and sometimes on the designer own judgement. The 'objective' information, $I_o$, can be various mathematical formulae for strength calculations. They are universally true and thus can be incorporated as parts of the computer program. The 'subjective' information, $I_s$ , are specifications from design codes, city regulations, etc. They vary with design environment and may need revisions from time to time. It is worthwhile, therefore, to bring them together into a replace-