COMPUTERS IN ENGINEERING 1985

VOLUME ONE

- Computer-Aided Design
- Computer-Aided Manufacturing
- Robotics
- Computer Graphics

COMPUTERS IN ENGINEERING 1985

Proceedings of the 1985 ASME International
Computers in Engineering Conference and Exhibition
August 4–8, 1985
Boston, Massachusetts

sponsored by

The Computer Engineering Division, ASME

Co-Editors

R. Raghavan
S. M. Rohde

Associate Editors

K. Ahluwalia
J. Lester
E. Patton
R. Rosenberg
T. Shoup
D. Whitney

VOLUME ONE

- Computer-Aided Design
- Computer-Aided Manufacturing
- Robotics
- Computer Graphics

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

Library of Congress Catalog Card Number 82-072306

Statement from By-Laws: The Society shall not be responsible for statements or opinions advanced in papers . . . or printed in its publications (B7.1.3)

FOREWORD

The papers in these three volumes were presented at the 1985 International Computers in Engineering Conference and Exhibit held in Boston, Massachusetts during August 1985. Since its beginning in 1980, the ASME Computers in Engineering conference has become a major international event bringing together experts from industry, government, and academia concerned with the ever evolving role of the computer in engineering.

This year's theme was "Expert Systems: A New Dimension in Computer Engineering", and the technical content of the conference focused on areas related to research, development and applications of computers in mechanical engineering with emphasis on expert systems. The scope of the conference included robotics, computer aided design and manufacturing, and computer aided engineering, as well as computers in education, simulation, concurrent computations, software and hardware design, optimization, graphics, and modeling. This year 65 sessions were held including panel and tutorial sessions, 264 papers were contributed and participants came from 25 countries. Keynote speakers were Marvin Denicoff, Raj Reddy, H. Bloom, and Nien-hua Chao.

This year's technical program was put together by the following individuals who deserve special credit for the excellent job that they did: K. S. Ahluwalia, J. Lester, E. M. Patton, R. C. Rosenberg, T. Shoup, D. E. Whitney. In addition we would like to acknowledge the efforts of the International Program Chairman — A. Seirig, the Exhibits Chairman — D. Dietrich, and the ASME staff for their support and help.

On behalf of the Computer Engineering Division and the Conference Committee, we thank the speakers, session organizers, chairpeople, and the authors for their contributions. We look forward with anticipation to the new challenges in the years ahead.

R. Raghavan, Conference Chairman
S. M. Rohde, Conference Technical Program Chairman
Co-chairmen, Editorial Committee

CONTENTS

THE ROLE OF TURNKEY CAD/CAM SYSTEMS IN THE DEVELOPMENT OF THE "GRAPHYSIS" CONCEPT

I. Zeid
Department of Mechanical Engineering
Northeastern University
Boston, Massachusetts

T. Bardasz
Computervision Corporation
Bedford, Massachusetts

ABSTRACT

Vendors of commercial turnkey CAD/CAM Systems have been focusing recently on Computer Aided Engineering (CAE) applications to help designers and analysts use these systems efficiently to meet specific design and analysis needs. An assessment of these activities indicates that two issues must be considered to achieve maximum benefits of the CAD/CAM technology. First, engineering community must use turnkey systems innovatively in their design and analysis tasks. Second, true integration between interactive graphics and traditional engineering analyses must be developed. This paper describes the development of a concept called "Graphysis" (Graphics and Analysis) that addresses these two issues. The concept has three levels of application: introductory, intermediate, and advanced. The concept statement, its applications, and its results are presented. A comparison with other existing concepts is outlined. The impact of the concept on Artificial Intelligence (AI) is assessed.

INTRODUCTION

The interface between the rapidly advancing CAD/CAM technology and various engineering applications has attracted the attention of engineers and researchers both in industry and universities (1-5). The major advantages of such interface between traditional engineering analyses and interactive graphics techniques can be summarized as follows:

1. Reduction of efforts in data preparation; this is greatly recognized in particular in the finite element area by developing pre-processors for mesh generation,
2. More effective display of the results; Results from various engineering analyses can be displayed graphically via efficient and well written post-processors. Among these processors are those developed for finite element and simulation packages.
3. Performance of parametric studies effectively; using a fully integrated interactive analysis/graphics program, a designer or analyst can investigate the "What If" question more efficiently by rapidly testing various cases and comparing the results on a graphics display.

The interface between engineering analyses and interactive graphics can be achieved on three types of hardware configurations. First, traditional main frame computers can be used as a driver for the engineering calculations and traditional graphics terminals can display the results. The shortcoming of this approach is that it always lacks a well structured and general database for further development or for other applications. Second, microcomputers offer good capabilities for analysis and graphics. A limitation of this approach may be the lack of space and the speed to carry extensive engineering applications. The third configuration is Turnkey CAD/CAM systems. These systems are typically super-minicomputer based systems and offer a general well developed database structure and management that can provide a good basis to carry engineering analyses. The computational power of these Turnkey CAD/CAM Systems can be employed to mix the traditional engineering analysis with the interactive computer graphics to develop a concept called the "Graphysis" (Graphics and Analysis) concept. This concept can be developed at the database level which makes it more efficient than other existing interfaces.

Survey of available Trunkey CAD/CAM Systems is beyond the scope of this paper. However, comparisons among few of them such as computervision (Designer V, CDS400, and CDS5000 Systems), Calma, and MCAUTO reveal that typical uses of these systems include five major levels. These levels are listed below in the order of the ease to use them:

1. Graphics applications; this is the most popular mode of using these CAD/CAM systems. Typically designers/analysts and draftsmen build 3D design models that usually serve the basis for drafting, numerical control, and/or finite element modeling purposes later.

2. Basic elementary engineering applications; mass property calculations and/or animation techniques are typical applications which are a spin off the geometric information generated from the previous level. The applications are usually achieved via

graphics commands offered by the CAD/CAM Systems and whose syntax is Similar to that of the graphics commands available on the System.

3. Elementary level of programming; this tool is mainly provided for generating family of parts that have the same topology but different dimensions.

4. Interactive programming languages; any one of these languages provides a typical user with statements for interactive graphics and others for calculations. The syntax for the graphics statements is very close to that of the graphics commands. Fortran language is also available on most systems. Computervision provides users with VARPRO2, PEP, NEWVAR, PASCAL, and FORTRAN-S languages. Calma provides users with DAL and FORTRAN while McAUTO makes GRIP available.

5. Graphics system programming; this is the highest level of programming a Turnkey CAD/CAM system can offer to a user. However, it requires considerable knowledge of the database structure and management of a particular system. Programs developed using this level are more efficient and usually require less user input than if they are developed using the interactive programming languages discussed in level 4.

Most of the current engineering applications and interfaces are achieved via the concept discussed in level 4 (1-5). A major common limitation of this level is that the interface between analysis and graphics does not occur at the database level. There are always internal translators that make the developed programs require longer time to run.

THE "GRAPHYSIS" CONCEPT

The concept is based on the fact that graphics databases of geometric models contain analysis information readily available to be harnessed for analysis and/or decision making (Artificial Intelligence). If more information is needed, the user must provide it via programming using either level 4 or 5 mentioned in the introduction section. Thus the major goal of the "graphysis" concept is to develop an optimum structure and management system of a new database called the "graphysis" database to handle the graphics and analysis information of a certain engineering model. One criterion to develop such a graphysis database is to input the analysis data simultaneously with the graphics data. On one extreme, this database can be the typical graphics database of the model provided by the CAD/CAM system if the analysis level is not sophisticated. On the other extreme, the whole database may have to be developed.

Analogous to the various models such as geometric, shading, and color models that form the heart of existing CAD/CAM Software, a model called the "graphysis" model associated with the concept is to be added. In addition, and similar to the different types of geometric (wireframes, surfaces, solids), shading (diffuse illumination, specific light sources, transparency), and color (RGB, CMY, YIQ, HSV, HLS) models, the graphysis model can be classified according to its implementation as follows:

1. Application Model

This represents level 2 described in the introduction. It is the most widely used model currently. The model consists of application packages provided by CAD/CAM vendors to perform certain engineering analysis. Mass property calculations are a typical example. The model uses the graphics database of a part to perform the analysis. No prior programming or database knowledge is required to use such a model. However, a knowledge of the analysis concepts such as the volume, mass, centroid, and inertia are necessary to both use the model effectively and interpret its results.

2. Interface Model

If the analysis is more involved and requires programming of formulas and equations or if an analysis package exists (such as finite element codes), then an interface between two databases must be formulated and developed. While the first of these two, the graphics database, has a well established structure and management system, the second, the analysis database, is not fully developed and currently represents the analysis program or package itself. The separation of the two databases, in most cases, results in having to write interfaces before and after carrying the analysis and/or repetition in the input data. Figure 1 illustrates the communication link between the two. Interface I is required before analysis to extract from the graphics database the appropriate information necessary to carry the analysis. Interface II does exactly the opposite and is used to display the analysis results. Due to these interfaces, the model is not always efficient and may be cumbersome to use.

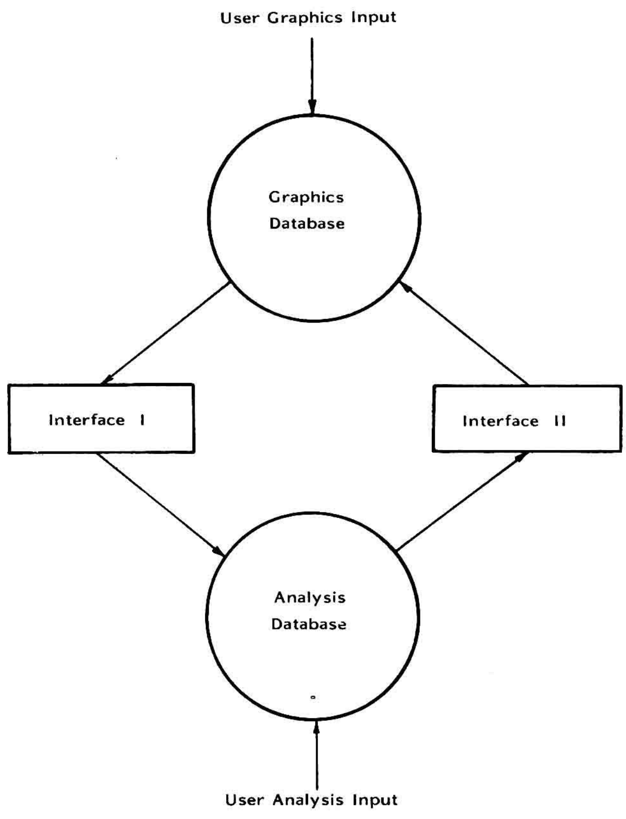


Figure 1 Communication Link for the Interface Model

3. Integrated Model

This model avoids the interfaces required by the previous model by adding the analysis information at the database level. This is accomplished by utilizing the current structure and management system of the graphics database and expands it to incorporate the analysis database, thus creating the "graphysis" database. If the graphics database structure proves to be inefficient, then a complete new structure of the "graphysis" database must be developed. The input required by the user for this model ("graphysis" input) is compact. Figure 2 shows three levels of utilizing a "graphysis" database. In addition to using it for graphics only, it could be utilized for analysis and/or graphysis. Once a "graphysis" database is built, the user can access it through an interface exactly similar to that to access a typical graphics database. In other words, "graphysis" commands could be developed with the same syntax as graphics commands. The modifiers for such commands include all the necessary information to perform the particular analysis. Menu capabilities could also serve as another effective user interface. If a considerable amount of time is needed to perform an analysis, the user interface should include batch processing capabilities. A better solution is to build "graphysis" hardware generators to speed up the execution of the analysis; thus maintaining the interactive nature of the model.

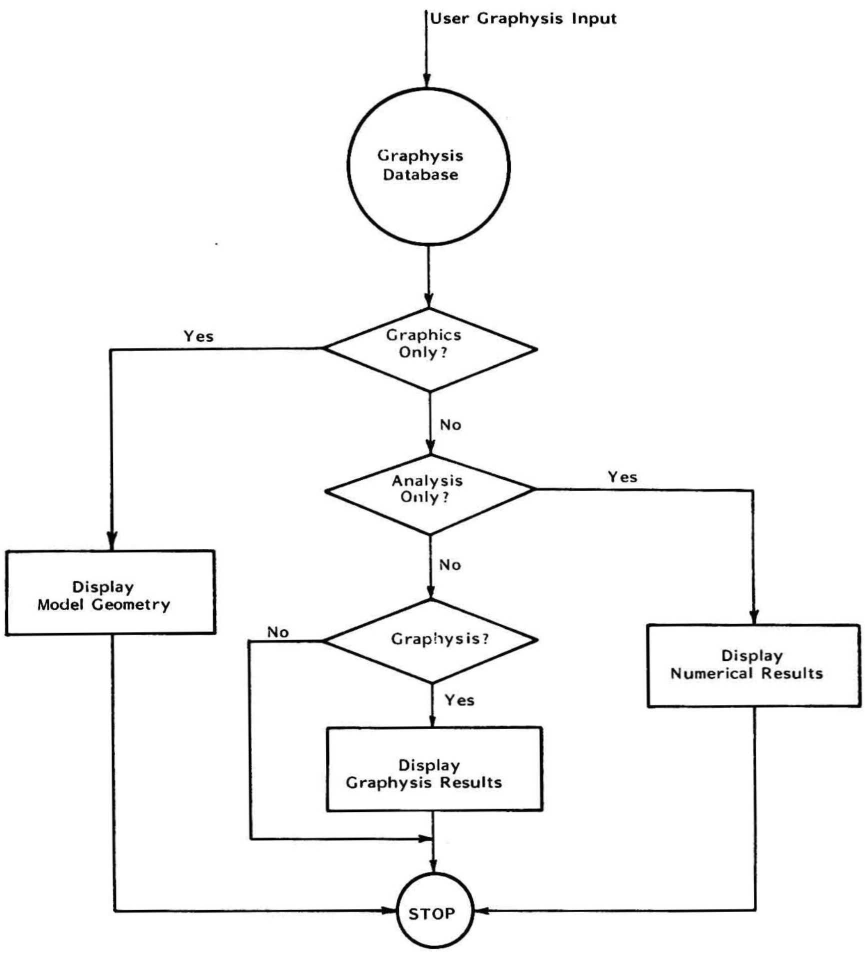


Figure 2 Proposed Utilization of a "Graphysis" Database

CONCEPT DEVELOPMENT

The development plan of the concept is directly related to the model best suited for the analysis. If it is an application model, particular features such as symmetry with respect to certain planes and/or uniform cross section in a specific direction must be taken into consideration during the concept development for such a model. These features are translated afterwards into graphics commands that help perform the analysis faster and more efficient. Such commands may utilize transformation (rotation, translation, and mirroring), duplication (copying), measure, and verify functions.

For an interface model, the analysis and the graphics are well defined and the knowledge of the programming languages to achieve the interface between the two is what is needed. These languages provide the user with access to the graphics information without getting involved with database structures and management systems.

The major element in the development of an integrated model is to decide on how the analysis data and results will be integrated with the graphics data and display. The following general rules can be used:

a) Analysis information that does not require graphics information can be input via modifiers within the "graphysis" commands or as separate commands. For example, if a beam analysis is to be performed, the load P and the modulus of elasticity E could be input on a computervision systems as:

#n# GEN DEF Pload xxxx MODE yyyy: Get data

OR

#n# INS PLOAD: Model value xxxx

#n# MODE : Model value yyyy

b) Analysis information that requires graphics information should not be input by the user and should be calculated directly from the model graphics by prompting the user to digitize the appropriate graphics items or entities. For the beam example, the moment of inertia of the beam cross section I can be found if the user digitizes the perimeter of such cross section.

c) Default values of the analysis parameters must be established and conveyed to the user.

d) If the analysis results are to be a graphics display, these graphics (a deflected beam shape) should be displayed automatically after the completion of the "graphysis" command by the user. If results are to be in the form of tables, then a separate command can be developed. For example,

#n# DISP RES

The DISP RES (Display Results) command can provide the user with modifiers to display beam displacements, moments, shear forces, and/or slopes for the beam example in the form of tables and/or diagrams.

CONCEPT IMPLEMENTATION

The implementation of the development related to the integrated model only is covered in this section. The implementation process results in creating a single (or a group of) "graphysis" command which can be executed by the user. Irrespective of the specific details of hardware and Software of various CAD/CAM systems, the following generic steps are required to generate a "graphysis" command:

1. Generate user interface: following the existing command syntax, the command structure and modifiers are added to the Software.
2. Extract data: If the command requires digitizes by the user, the appropriate geometrical information (such as the perimeter in case of the beam) is extracted from the "graphysis" database to be used in the next step.
3. Perform calculations: the "graphysis" information generated in steps 1 and 2 are used to generate the desired results.
4. Display results: results will be extracted from the database to be displayed in the desired format.

As a case of illustration, the above steps are equivalent to the command environment available on the Computervision (CV) System. More specifically, to create a new command on the CV System requires (6-8):

1. Verb-Noun Processor: which is ready to receive a CADS command from the user. The "graphysis" command must be inserted into the verb-noun table to make it legal.
2. Modifier Processor: identifies the legal modifiers that accompany the new command. The modifiers either direct the computer to a specific part of the data processor to be used or to input values to be used in data processing.
3. Get-Data Processor: accepts input taken from graphics entities displayed by digitizing them or the input of explicit coordinates or real values to be used as input to the data processor. A read from model database may be performed by this processor.
4. Data Processor: uses the data from the modifiers and get data to make calculations.
5. Graphics Processor: takes the results from the data processor after being written to the model database and creates their corresponding graphics display.

Users of other commercial systems should be able to find similar steps on their respective systems within the guidelines of the generic steps mentioned earlier.

APPLICATIONS

Three "graphysis" applications are performed on the computervision CAD/CAM system to illustrate and demonstrate the effectiveness and applicability of the "graphysis" concept. The first two applications fall under the application model while the third represents an integrated model. Examples illustrating the inteface model are not included but are referred to when appropriate.

The first application is simple in nature yet very powerful when the geometry of the model involved becomes complex. In this application the mass property calculations of a symmetric model are calculated. The "graphysis" concept in this case can be applied as follows. If the symmetry is used to construct half of the model geometry and then mirror it about the plane of symmetry to obtain the full model, the same observation is applicable to its mass properties and should be used to calculate them. The basic mechanics principles (9,10) supports this observation. Referring to Figure 3, the mass properties of the model right and left halves are the same due to its symmetry about the YZ plane. If these properties are added, the centroid of the total model (which lies on the plane of symmetry) and its inertial properties are given by (9,10):

$$X_c = \left(\sum_{i=1}^{2} Mi\ Xi \right)/(M_1 + M_2) \quad (1)$$

$$Y_c = \left(\sum_{i=1}^{2} Mi\ Yi \right)/(M_1 + M_2) \quad (2)$$

$$Z_c = \left(\sum_{i=1}^{2} Mi\ Zi \right)/(M_1 + M_2) \quad (3)$$

$$I_z = \int_{M_1} (X^2+Y^2)\ dm + \int_{M_2} (X^2+Y^2) dm \quad (4)$$

Similar formulas can be written for I_x and I_y as well as the products of inertia. Thus, when one half of the model geometry is used in the calculations, 50% of the user digitizes are saved which in turn saves the corresponding computer time associated with the get-data processor. The properties of the second half are obtained by mirroring those of the first half.

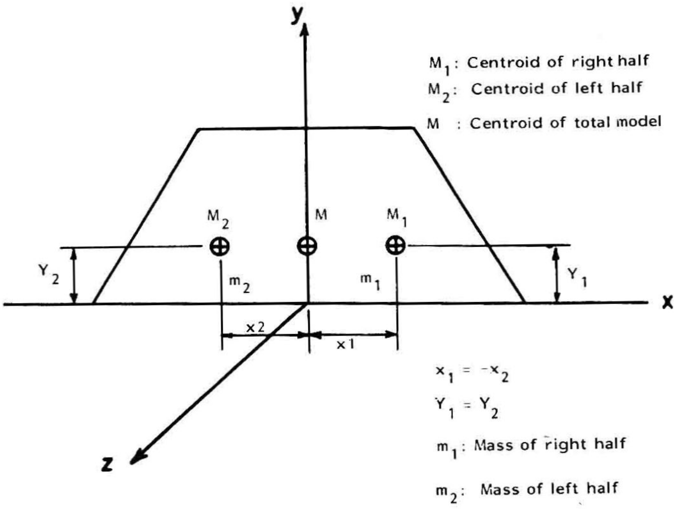


Figure 3 Mass Properties of a Typical Symmetric Model

to give the locus of E in space. Using the Instantaneous center method, the following velocity equations can be written (10)

$$W_{BC} = W_1\ L_1/BI \quad (5)$$

$$V_E = W_{BC}\ EI \quad (6)$$

$$W_{CD} = W_{BC}\ CI/L_3 \quad (7)$$

where BI, CI, EI are distances measured from the instantaneous center I of BC to points B, C, and E respectively. I is located graphically for each angle θ and the lengths are measured using the MEASURE DISTANCE command. The direction of the velocity V_E can be found by inserting a tangent to the B-spline locus at the appropriate location.

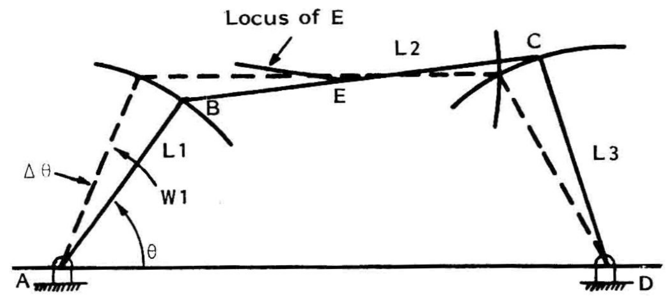


Figure 4 Geometric Model for Kinematic Analysis of a Four-Bar Linkage

The objective of the second application is to illustrate how information in a model graphics database can be extracted and employed to perform a required analysis. Figure 4 shows a four-bar linkage whose kinematic analysis is to be performed. While a more comprehensive analysis using the complex numbers theory (11) can be performed, the locus and velocity calculations of Point E, the center of the coupler link BC, using the model graphics database are presented here. The user starts by constructing various configurations of the mechanism using a certain angle increment $\Delta\theta$. For any one configuration, the input link AB can be input in polar coordinates. Point C can be found as the intersection of two loci : the first is an arc or a circle with center at D and radius L3 and the second is also an arc or a circle with center at B and radius L2. A point is inserted at E for each configuration to record its location. Each point has a time value associated with it. The resulting points can be connected with a general curve such as a B-Spline curve

In the final application, a "graphysis" database of a cantilever beam is built. This database contains geometric information such as the beam length and its cross section geometry as well as analysis information such as load type (point or distributed), material type (Steel or Aluminum), and cross section type (rectangular, triangular, tubular, or circular). Given a cantilever beam loaded at its free end with a point load P (Figure 5), the deflection equation is given by (12):

$$PX^2(X-3L)/6EI \quad (8)$$

A new command is created to give the user an access to the new database by translating the required analysis information into appropriate analysis modifiers. Deflections at various locations along the beam length are calculated via eqn (8) and the resulting points are connected by a B-Spline curve to give the deflected beam shape. Other information pertinent to the analysis can be extracted from the database if needed.

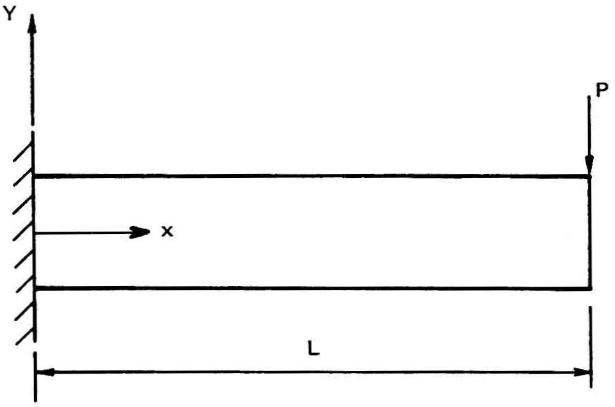


Figure 5 Coordinate System for Beam Deflection

RESULTS AND DISCUSSIONS

The three examples presented in the previous section were performed on the computervision CAD/CAM System. Figure 6 shows the mass points of various components of a typical model. If M_1 and M_2 of the left half were not mirrored, unnecessary time and effort would have to be spent especially if the XSEC modifier is needed.

Figure 7 shows locus of point E that was generated for $\Delta\theta$ of 10 degrees. In order to extract the analysis information, the graphics commands VERIFY ENTITY and MEASURE DISTANCE were used.

The command GENERATE DEFLECTION was created to implement the beam analysis. The syntax is:

#n# GENERATE DEFLECTION MATERIAL STEEL LOAD 1000
TYPE 1 XSECTION RECTANGULAR: Model Ent d_1 d_2 d_3

where type 1 refer to a point load and digitizes d_1, d_2, and d_3 are digitizes shown in Figure 8. The command displays the deflected beam shape. If this example was to be performed using the interface model, the user would have to write a VARPRO2 program that could call FORTRAN subroutines.

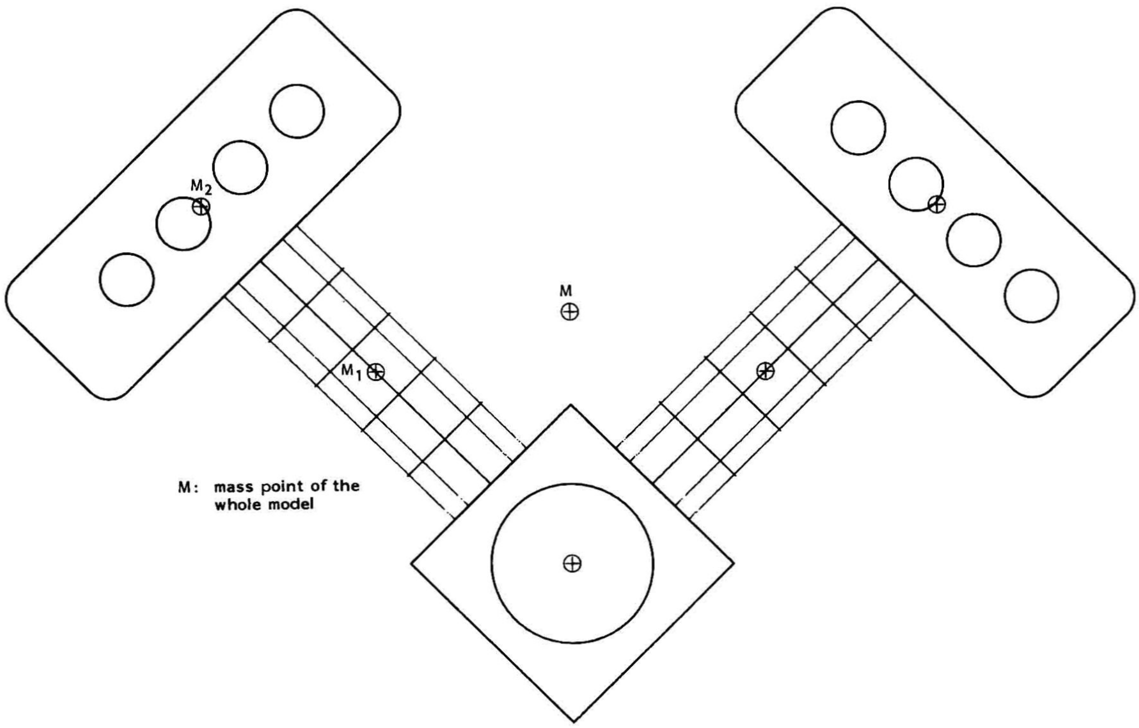


Figure 6.a Mass Points of a Symmetric Model; Front View

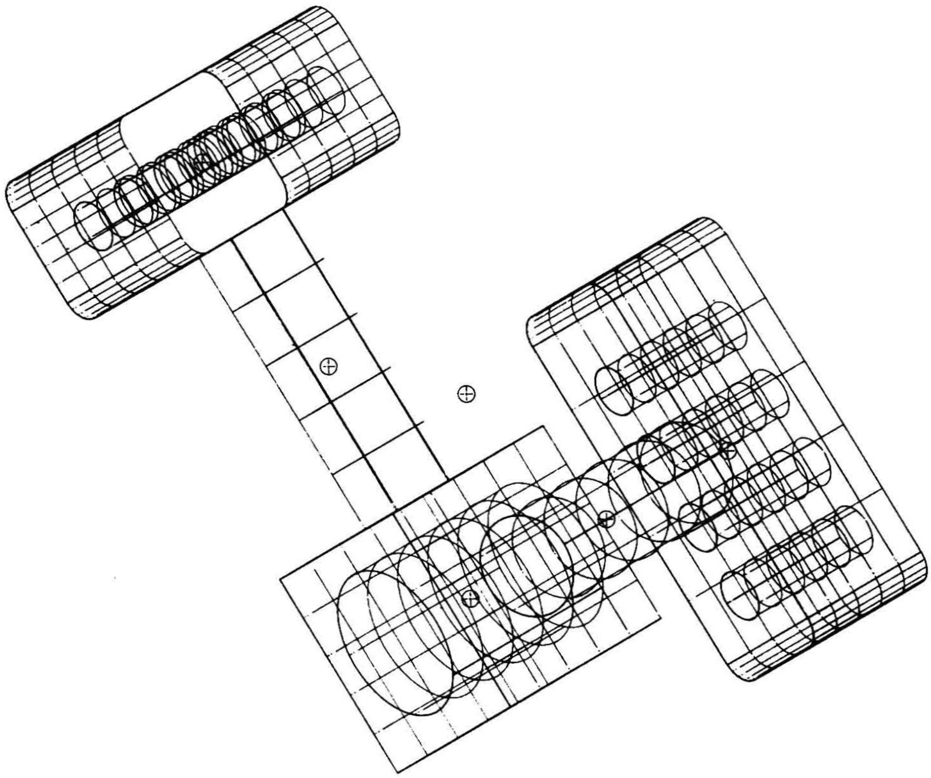
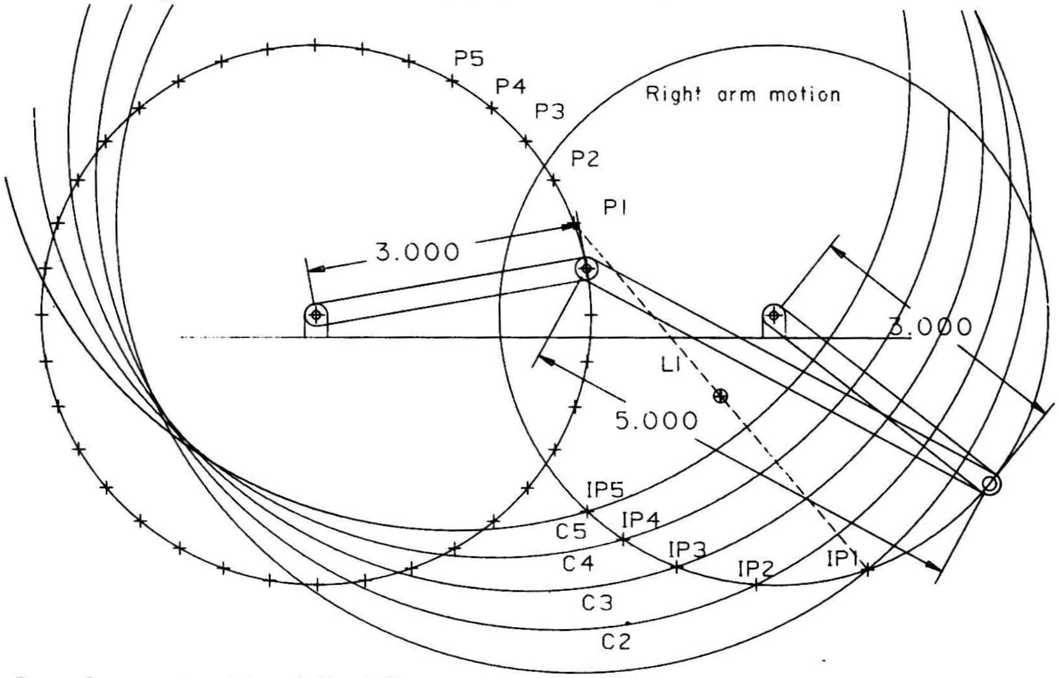


Figure 6.b Mass Points of a Symmetric Mode; Isometric View



Pn = Some set points of the left arm.
Cn = The range of motion of the middle arm.
IPn = A possible set of the locations for the "uncontrolled" end of the middle arm
L1 = The position of the middle arm when the left arm is fixed at pi

Figure 7.a Generation of Various Locations of Point E

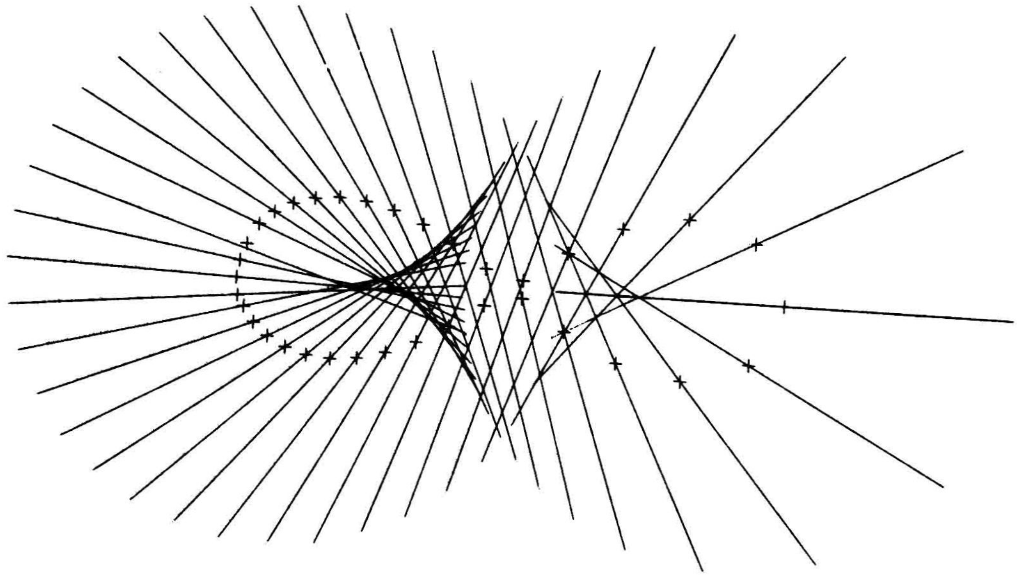


Figure 7.b Points Defining Locus of Point E

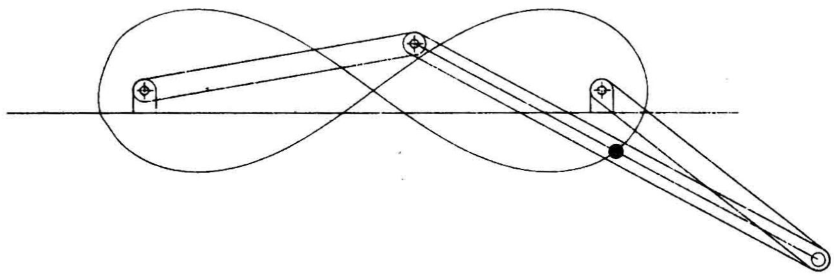


Figure 7.c Locus of Point E in Space

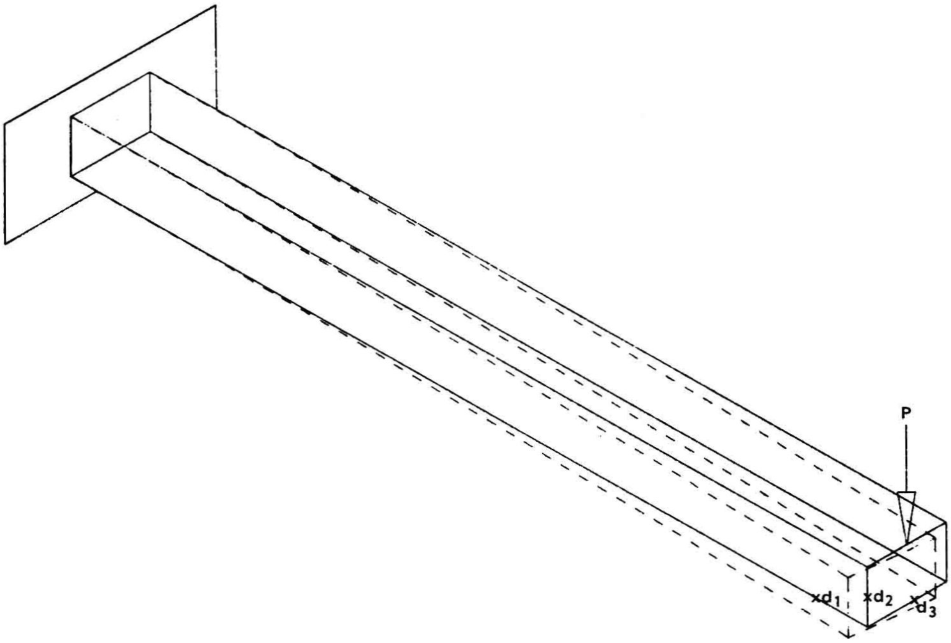


Figure 8.a Beam Deflection; Required Digitizes for "Graphysis" Command

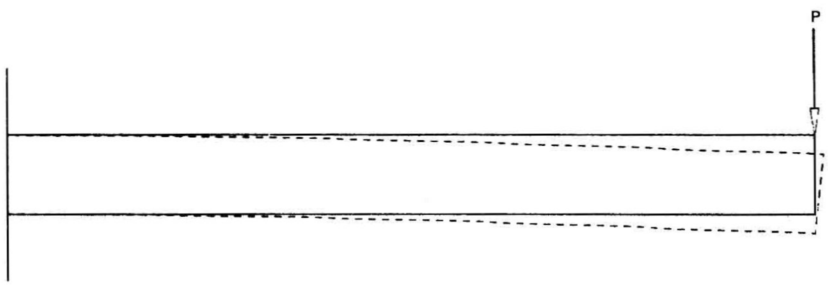


Figure 8.b Beam Deflection; Front View