Series 05 Aerospace Structures and Computional Mechanics 02

User's Manual for the Computer Program Cufus

Quick Design Procedure for a CUt-out in a FUSelage Version 1.0

M.E. Heerschap

Delft University Press

User's Manual for the Computer Program Cufus

Quick Design Procedure for a CUt-out in a FUSelage Version 1.0

Series 05: Aerospace Structures and Computional Mechanics 02

User's Manual for the Computer Program Cufus

Quick Design Procedure for a CUt-out in a FUSelage Version 1.0

M.E. Heerschap

Delft University Press / 1997

Published and distributed by:

Delft University Press
Mekelweg 4
2628 CD Delft
The Netherlands
Telephone +31 (0)15 278 32 54
Fax +31 (0)15 278 16 61
e-mail: DUP@DUP.TUDelft.NL

by order of:

Faculty of Aerospace Engineering
Delft University of Technology
Kluyverweg 1
P.O. Box 5058
2600 GB Delft
The Netherlands
Telephone +31 (0)15 278 14 55
Fax +31 (0)15 278 18 22
e-mail: Secretariaat@LR.TUDelft.NL
website: http://www.lr.tudelft.nl

Cover: Aerospace Design Studio, 66.5 x 45.5 cm, by:
Fer Hakkaart, Dullenbakkersteeg 3, 2312 HP Leiden, The Netherlands
Tel. +31 (0)71 512 67 25

ISBN 90-407-1584-X

Printed in The Netherlands

Contents

APPENDICES

Acknowledgement:

The work presented here is part of a research project carried out with the financial assistance and technical collaboration of Fokker Aircraft BV. The author wishes to thank Fokker for their support. The author also wishes to thank Prof. Dr. A. Rothwell and Prof. Ir. P.A. van der Schee, both of the Faculty of Aerospace Engineering, Delft University of Technology, Ir. C. Schimmel, and Ir. M.K. Khoudja, Fokker Aircraft B.V., for helpful discussions.

This User's Manual is intended for Cufus version 1.0, PATRAN version 2.5 and NASTRAN version 67.5. Almost all parts of the program Cufus also function properly using PATRAN 2.5 in combination with NASTRAN version 68, version 67.5 and version 67 Release 2.

1 Introduction

Cut-outs are unavoidable in all pressurized aircraft fuselages. Especially large cut-outs, like those for doors, remain very fatigue sensitive. This fatigue sensitivity is due not only to the fact that the cut-out causes a large stress concentration, but also due to the use of the door in service. This increases considerably the chance for accidental damage. When a small dent or other damage has occurred, a fatigue crack is only a matter of time. Fatigue damage can endanger safety, or at the very least leads to expensive repairs. For these reasons stress levels must be kept below carefully defined maximum values. In spite of adding all kinds of reinforcement, full scale fatigue tests often demonstrate fatigue cracking in the corner of the cut-outs. This is partly due to the fact that the methods of structural analysis used in the (initial) design of the cut-out are not very accurate, possibly because of the assumptions necessary to perform these calculations by hand [1, 2, 3]. When a more detailed finite element calculation can be performed, special attention must be paid to the area near the corner of the cut-out, because of the very high stress gradient at that location. Furthermore, it is common practice in aircraft design that some aircraft must be modified according to customers' wishes. The structural items frequently modified are the doors and their location in a pressurized fuselage. This kind of modification must, in general, be re-designed in a very short period of time. It is clear that there exists a need for a design system which offers a quick design cycle and an accurate calculation of the stress levels occurring in the structure.

Finite element methods are generally accepted as analysis tool for structural design. The design procedure is normally based on (a large number of) finite element calculations with resizing, until the desired goal is achieved. In its simplest form the design procedure consists of adding material in the highly stressed regions until all stresses are below a certain acceptable level. When adding material the assumption is normally made that the structure behaves like a statically determinate structure. This implies that the stress varies linearly with the inverse of the cross-sectional area. Furthermore, it is assumed that a change in a structural member only affects the structural member itself. It will be clear that these assumptions do not hold in a statically indeterminate structure such as a fuselage. The re-design of the structure could be greatly improved if so-called sensitivity data were available. These data describe the change in structural response (stress, strain, displacement, etc.) in the complete structure due to a change in one of the properties of a particular structural member. In this way the designer can make a more accurate prediction of the behavior of the structure when some changes are made. As illustration of the usefulness of sensitivity data, it will be obvious that adding material will reduce the stress locally but due to the changed load distribution stresses might be increased in another part of the structure. Prediction of the structural response in this way can improve the efficiency of the design procedure considerably, because less design iterations are needed.

A further step is the use of structural optimization. Optimization employs the same sensitivity data and steers the design towards some objective while satisfying the chosen constraints. The task of gradually improving the design is now performed by the computer instead of the designer, in this way saving much time. In general it can be stated that use of sensitivity data, whether or not combined with an optimization, greatly improves the effectiveness of the finite element method in design.

Most finite element codes offer the means to perform both a geometrically linear and a geometrically non-linear calculation. Some also offer options to perform structural optimization and/or a sensitivity analysis. The "ordinary" finite element calculations are nowadays well accepted in industry, but the use of a sensitivity analysis and optimization are still treated with some suspicion, in spite of the advantages stated in the previous section. Partly this is due to lack of accurate loads in an early stage in the design, and to lack of proper means of pre- and post-processing the data generated by a sensitivity analysis or optimization. The first problem cannot be solved here, but because the loads and the design are related to each other to some extent, a fast design method will assist in a fast determination of the loads as well. The problem of suitable means to post-process the data generated by a sensitivity analysis or optimization is solved with the proposed design system. In particular, post-processing of a sensitivity analysis requires a well considered, graphical presentation because of the huge amount of data generated. The same could be said for the structural optimization but here is also the danger of arriving at a local optimum, or even divergence, both at the expense of a large amount of computing time. This has prevented acceptance by industry. Still it should be noted that a properly used optimization makes maximum use of the generated sensitivity data, and in general arrives at a better optimum, or at least it will reach an optimum faster, than can be achieved by an "optimization by hand".

This manual describes a design system offering the means of performing finite element analysis, sensitivity analysis and structural optimization in an easy, user-friendly way. The system provides several means to post-process sensitivity data, including use of a graphical display of these data, and 'what-if' studies. The finite element model for the analysis is easily generated and pre-processed by the design system. To achieve the goals of the design system, a menu-driven, graphical environment is used. Contrary to Santos et al. [4, 5] the design system is set up as a special purpose design tool. This special purpose aspect focuses mainly on the model generation.

Maximum use is made of commercially available software. Furthermore, it should be noted that all software written for use with this design system is in addition to the available software. That means that no modification of the source code of the commercial software is needed; the design system is aimed of making the best of the combination PATRAN/NASTRAN. Most of the extra computer code is written in PCL, which is a structured programming language available in PATRAN [6]. The main reason for programming in PCL is to make use of PATRAN's graphical user interface for the design system. Furthermore PATRAN offers the possibility of creating a user-defined menu, which is a powerful tool to create a user-friendly design system.

Only limited knowledge of PATRAN is required to operate the design system Cufus. However, a small number of tasks are performed by PATRAN without the help of the user interface provided by Cufus. These tasks are also described in this manual. If the user is familiar with PATRAN, this will simplify the use of Cufus. If the user wants to program further in Cufus, detailed knowledge of PATRAN and PCL is required [6, 7].

It must be emphasized that this User's Manual is no replacement for the PATRAN User's Manual [7]. The items treated in this manual are all features added to PATRAN, the standard PATRAN options are all covered thoroughly in the PATRAN User's Manual.

2 Global Program Description

2.1 Operational Requirements

The requirements discussed in this section are only appropriate for use of the program at the T.U. Delft Faculty of Aerospace Engineering, working on one of the SUN stations and the multi processor computer of the C-group. In case one wants to work at a different site, these requirements are not in general valid. How to meet the requirements is described in appendix A. This appendix also describes how to start PATRAN, Cufus and MSC/NASTRAN.

2.2 Program Philosophy

In this section some of the "strategic" choices, made during the development of the program Cufus are described. First of all it is necessary to review the requirements for the program. It is required to generate in a quick way a detailed model of an aircraft fuselage with a cut-out for a door. The shape of the fuselage is always single or double circular (i.e. a simple circular shell or a double-bubble fuselage shell). This model is the basis for a linear static, a non-linear static, a sensitivity and an optimization finite element analysis. All finite element analyses must be performed with MSC/NASTRAN. The program must be able to handle various load cases and boundary conditions.

These requirements result in a number of limitations on the program. One of the most obvious limitations is the restriction that the shape of the fuselage must be single or double circular. In case this is considered to be a major disadvantage, only limited re-programming would be needed, thanks to the modular structure of the program. Sufficient information is provided by the Programmer's Manual [8].

During the initial model generation a certain numbering of the geometric entities took place. In combination with the hard limits on the maximum number of geometric entities which are allowed by PATRAN, this limits the number of frames and stringers which can be used in the finite element model. These limitations are now: maximum number of frames less than or equal to 10, and maximum number of stringers less than or equal to 98. It should be noted that these values are only advised limits, small deviations from the mentioned limits may occur rarely.

Other limitations are imposed by MSC/NASTRAN. In the version used at this moment (version 67.5) it is only possible to perform a sensitivity analysis and a structural optimization for a linear analysis. It is also impossible to combine the structural responses from different load cases with mutual exclusive boundary conditions in one structural optimization job. This is caused by the fact that the solution sequence for this type of analysis is a so-called super element analysis. MSC has stated that the last mentioned limitation will be eliminated in MSC/NASTRAN version 68. Most parts of the design system Cufus are already tested with MSC/NASTRAN version 67R2 and version 68. However, the extra possibilities provided by especially version 68 are not yet utilized.

These MSC/NASTRAN limitations mean that no sensitivity data is available for a non-linear analysis implying that if a "what-if" analysis is performed on the basis of a non-linear static analysis the sensitivity data of a linear static analysis should be used. Therefore, in case of highly non-linear behaviour the results coming from such analysis should be handled with care. Fortunately the influence of the geometrical non-linearity at "normal" fatigue stress levels in a pressurized fuselage (i.e. < 100 MPa) is in most cases rather limited.

2.3 General Description of the Program

Cufus consists globally of three parts, namely: 1) initial model generation, 2) model editing and definition, 3) post-processing. The first part of the program is the initial model generation. This part involves entering some key dimensions, or reading these dimensions from an input file. This initial modelling stage is meant as a start for the user. An initial mesh is generated on top of the geometry model. This mesh, however, will very likely be modified in the next stage. Properties are not assigned in the initial modelling stage. The key dimensions, which should be entered or read from file, are drawn in the figures 1a and 1b, and the format of the file is given in appendix C. The global system set up of Cufus is given in figure 2. This figure also refers to sections in which the subjects are discussed in detail.

During the initial model generation information is taken from the normal PATRAN data base and stored in a local Cufus data base. This local data base is normally stored in the core memory of the computer and accelerates the interactive modelling task considerably by eliminating the time consuming repeated disk access. Filtering the information from the

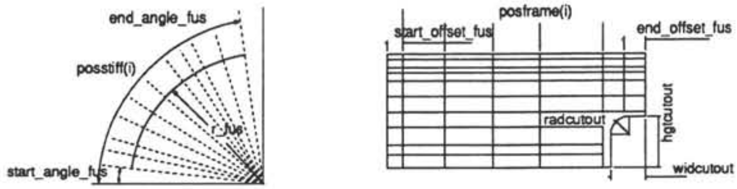


Figure 1a: Definition of some dimensions of the fuselage and the cut-out, for a simple cross-section.

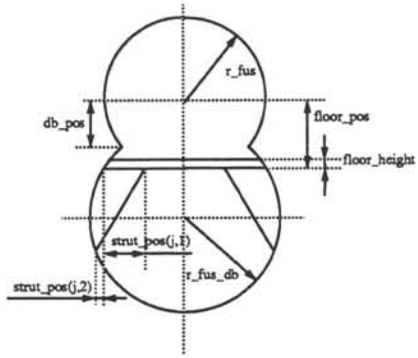


Figure 1b: Definition of some dimensions of the fuselage for a double-bubble fuselage with floor.

PATRAN data base into the local data base takes some time (in the order of a few to 20 minutes). This can be reason enough to run PATRAN in batch mode. This saves some time because no graphics are displayed while running in the batch mode. However, after the initial model generation the user has to quit PATRAN in the batch mode and switch to PATRAN in the X-windows mode, which takes probably as much time as was saved by running it in the batch mode. The total time needed by the initial modelling phase will depend on the size the model, but will be typically in the range of 5 to 20 minutes.

The initial model generation is dedicated completely to creating a part of a fuselage with a cut-out. This implies that if the program is used for creating similar problems, this part of the program should be rewritten. The two other stages, the model editing and definition, and the post-

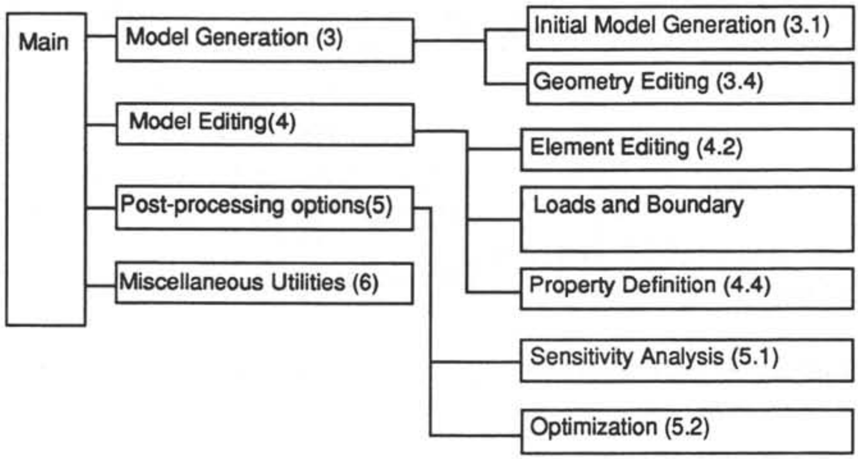


Figure 2: Global structure of Cufus. The numbers between the brackets refer to the section numbers.

processing stage, can be copied almost unchanged. Detailed information is provided by the Programmer's Manual [8].

After the initial model generation, additional modelling can take place in the model editing and definition phase. This second stage is fully interactive and controlled by using the menu. In this section only the global capabilities are described is how to use these capabilities will be treated extensively in the following sections.

In the phase of model editing and definition limited changes can be made to the geometry. Also mesh adaption can take place. Furthermore properties can be assigned, application of loads and boundary conditions takes place in this phase, and finally optional assignment of design variables and definition of constraints is part of the model editing stage. Because the standard PATRAN-NASTRAN interface does not generate a MSC/NASTRAN Case Control Deck, the generation of such a deck is also included in this stage. Besides the absence of the Case Control Deck with the PATRAN-NASTRAN interface, the Bulk Data Deck is also not complete for a sensitivity analysis or structural optimization. Cufus will supply the necessary additions to the Bulk Data Deck during this phase.

The post-processing possibilities are basically extensions of the "normal" post-processing options. By "normal" post-processing options is meant the options for displaying stresses, strains, displacements, etc. Cufus provides some options to display sensitivity data in various ways and to post-process optimization data as well. The "what-if" studies, i.e. predictions of the stresses based on extrapolation of the first derivatives of the structural responses to the design variables, are in fact extended post-processing options based on the sensitivity data.

As well as post-processing sensitivity data there is also an option for post-processing a structural optimization. This consists of the visualization of the history of some important parameters of the optimization.

In particular the post-processing options are easy to transfer to another dedicated design tool or as stand alone program for post-processing MSC/NASTRAN jobs. Detailed information is provided in the Programmer's Manual [8].

The execution of any PCL or PATRAN function can be stopped by typing Control-C. It is possible that typing Control-C must be repeated several times, depending on the type of function to be stopped. However, this manner of stopping a function is ***not*** recommended because the data could be damaged. This is due to interrupting the function at the current position, which does ***not*** undo the changes already made in the particular function. Furthermore this type of data could result in an invalid data structure, which causes the loss of all changes made in the design session.

2.4 Using the Menu System

Cufus has an extended menu system. All functions are, however, also usable at the command prompt, but in that case some knowledge is required of the names of the PCL functions and the arguments of these functions. All these function names and arguments are extensively described in the Programmer's Manual [8]. The menu system will be in almost all cases more user-friendly and more efficient than using the PCL commands. The menu system is (for the biggest part) stored in the file called patuser.emn. Each menu item relates to a sub-menu, a PCL command or a PATRAN command. This makes the system easy to use and easy to modify. A description for creating and/or modifying the user defined menu can be found in appendix B, or more extensively in PATRAN Command Language (PCL) [6], chapter 9. The user is free to add options to the existing menu system without changing the original functionality of the menu system.

The complete PATRAN screen is shown in figure 3a. Here different menu parts can be distinguished. The alpha area is the area where commands can be typed. To enter commands the text in the menu on the right of the screen will disappear, the cursor will change shape and the alpha area shows an input prompt (>) at the bottom line of the alpha area, as shown in figure 3b.

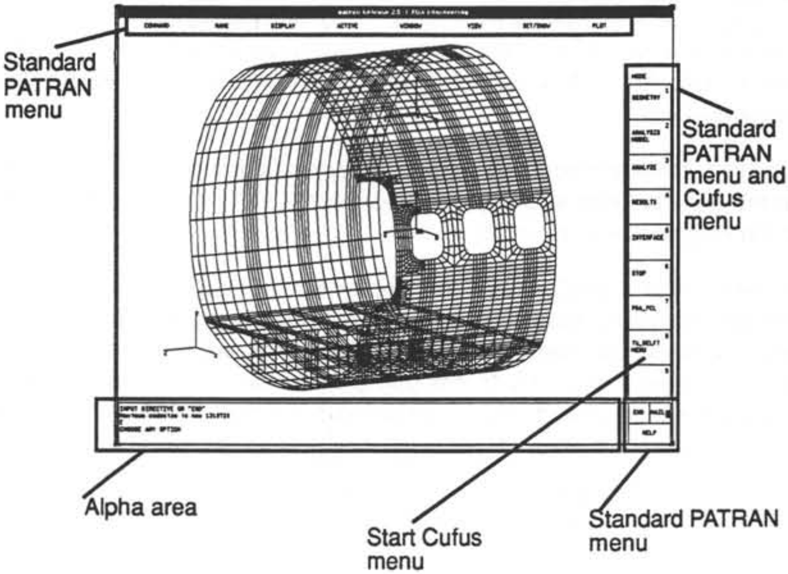


Figure 3a: Patran window showing several menu areas.

Figure 3b: The alpha area in command mode.

The menu options can be invoked by typing the number which occurs in the menu cell, or by clicking the option with the mouse cursor. The menu can only be clicked with the square cursor, ⊡. If the cursor looks circular, ⊘, PATRAN expects keyboard data to be entered. Travelling back through the menu system can be done by clicking the "end" option in the lower right corner of the PATRAN window. Another possibility to travel back is to use the ninth option of each menu presented on the right of the PATRAN window. Both options are sketched in figure 4.

The complete menu system is drawn in figure B1, appendix B. This menu system is partly stored in the file patuser.emn, as stated before, and partly written in PCL-code. The reason for including a part of the menu system in the program code is that it offers the opportunity to make the menu structure conditional. For the user it is possible to modify both parts of the menu, although modifying the part which is stored in the file patser.emn is easier then changing the menu part included in the code. The file patuser.emn can be used for personal extensions to the menu system. The different parts are indicated in figure B1, appendix B.

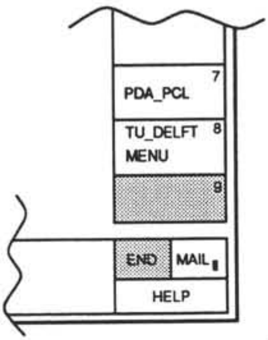


Figure 4: The two possible ways to travel back in the menu system.

2.5 Intermediate Back-Ups

The primary goal of this section is to emphasize the importance of making intermediate back-ups of the finite element model and the design model during the session. It happens frequently that changes made to the model do not have the effect the designer hoped for, or for example the design variables are chosen in a somewhat unfavourable way. Therefore it is recommended to save the model and the Cufus data on a regular basis during the design session. Furthermore this prevents a complete data loss in case Cufus or PATRAN should crash.

This section will show only briefly how to back up and restore a design session. In one of the following chapters, these matters will be discussed in more detail, see section 6.1. The reason why it is also mentioned in this section is to confront the reader at an early stage with the intermediate back-ups and to emphasize the vital importance of back-ups.

Attention should be paid to the fact that there are ***two*** data bases to save. The first one is the regular PATRAN finite element model, and the second one is the Cufus data base. The Cufus data base consists of information concerning mesh adaption, model topology, design variable definition and constraint data. It is important to keep track of which PATRAN data bases are linked to a particular Cufus data base. If one of the data bases is missing, this means a total data loss.

The back up of the PATRAN data base can be done via the menu selection shown in figure 5. The recovery of a PATRAN data base is performed by clicking the 'Input Model' option shown in figure 5. Questions by PATRAN in the alpha-area during writing the standard PATRAN data base should all be answered with "yes". The question asked during recovery of a standard PATRAN data base about the offset of the node identification numbers should be answered with "no". The PATRAN User's manual treats the subject of writing and reading neutral files in detail [7].

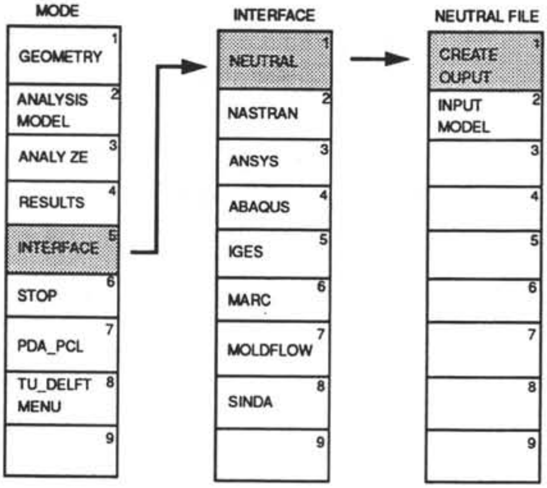


Figure 5: Saving the standard PATRAN data base

The way in which a Cufus data base is saved is shown in figure 6. To load an old Cufus re-start file, click 'Re-start' in this menu. A description of the re-start file can be found in appendix C.

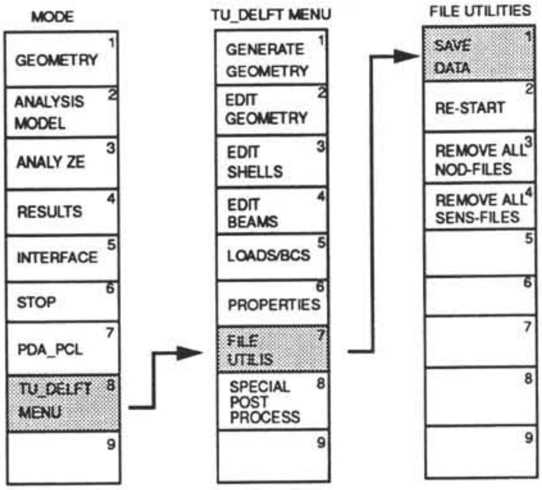


Figure 6: Saving the Cufus data base.

3 Model Generation

3.1 Initial Model Generation

As stated in the previous section, the initial modelling cannot be regarded as very interactive. The user can make a choice between entering the key dimensions by hand or reading these dimensions from file. The program needs to know the radius (or radii) of the fuselage. Cufus is restricted to circular shells and double-bubble fuselages. If the section to be generated is a 180 or 360 degrees section a floor can also be generated. Furthermore the number of stiffeners and frames and their positions should be entered.

The dimensions of the cut-out are restricted to rectangular cut-outs with rounded corners (door-shape) or to elliptical[1] shapes. Therefore it is only necessary to enter the height, the width, the position and the radius of the corners of the cut-out for the rounded rectangular cut-out shape. If the finite element model of the fuselage does not include the full height of the door (options A and B, figure 7), it is assumed that the horizontal axis of symmetry of the door lies on the start, in circumferential direction, of the fuselage.

Since it is also possible that sections of the fuselage can be modelled which span more than the full height of the door, an extra variable is required. This variable is used for entering the position of the door. this variable is the distance in the circumferential direction from the start of the fuselage ($\varphi = 0$ degrees, see figure 18) to the bottom of the door. A positive value means that the bottom of the door is above the start of the fuselage. This value must also be set for fuselages which span less than the full height of the door.

It is up to the user which part of the fuselage should be analysed. It is possible to give the initial and final angle of the fuselage. This feature can be useful if the internal pressure is the only load case to be analysed. In that case there is no need to model up to 90 degrees of the fuselage, but it is possible to reduce the size of the finite element model (depending on the size of the cut-out and the radius of the fuselage). If there are load types other than only internal pressure to

1. It will be obvious that elliptical shapes also include circular shapes.

be analysed, the user is obliged to use 90 degrees or a multiple of 90 degrees of the fuselage. The possible options are show in figure 7.

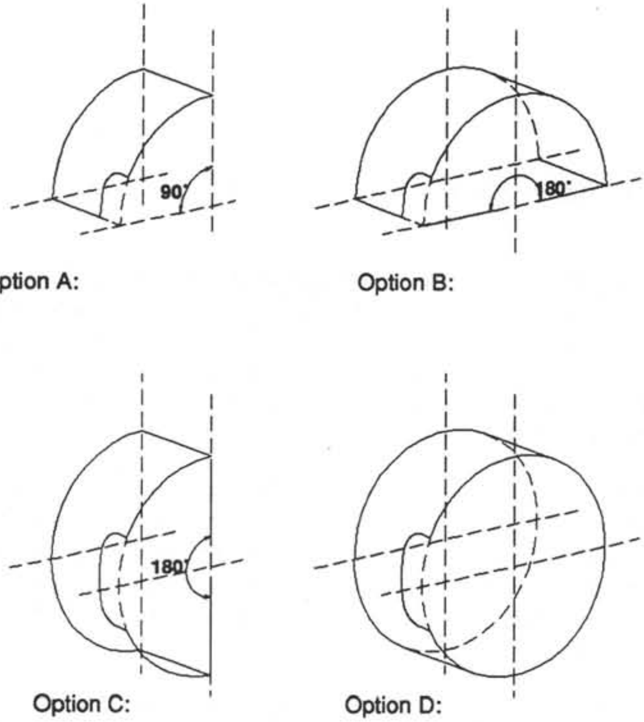


Figure 7: The possible fuselage sections.

The program will recognize the chosen modelling option and will adjust the boundary conditions for that size of fuselage shell. A 360 degree model and a 180 degree model (option C, figure 7) require a slightly different input, which will be discussed in section 3.2.

The last items to be entered are a few different offsets. These offsets make it possible to start and to end the fuselage, both in the axial direction and in the circumferential direction, at a different position to the first or last frame or stringer.

For a 360 degree model or a 180 degree model (option C, figure 7) some extra input is required for generating a floor and/or a double-bubble fuselage, see section 3.2). Even if these options are not wanted, some data concerning them have to be included in the input file.

Some of the dimensions discussed in this section are drawn in the figures 1a and 1b, see page 7. The format and complete contents of the input file are described in appendix C.

When PATRAN is started and Cufus is correctly loaded a menu is presented on the right side of the window, figure 3a and 8. By putting the mouse cursor on one the options and pressing the left mouse button, an option is selected. When the option TU Delft menu is chosen from the PATRAN root menu, figure 8, then the root menu of Cufus is presented, figure 9. This figure also shows the sub-menu for initial modelling which appears when the hatched option from the Cufus main menu is chosen. Before starting the generation of the fuselage it is useful to modify the

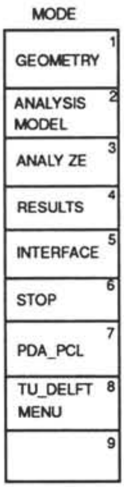


Figure 8: The PATRAN root menu

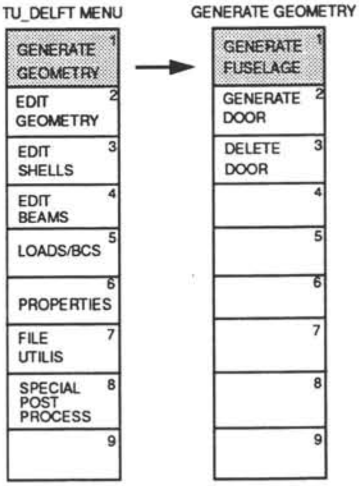


Figure 9: The Cufus root menu and model generation menu.

viewing angles of the finite element model (see chapter 4, ref. [7]). The default view is a front view and therefore showing only a very few details of the model generation. As soon as the initial modelling terminates, the viewing angles can be modified again.

If the user has picked one of the options presented in the sub-menu "Generate Geometry" some data has to be entered via the keyboard. For entering this type of data PATRAN uses the so-called "alpha-area". This is the rectangular box on the bottom of the PATRAN window, see figure 3b. The fact that the program asks for data from the keyboard is also made clear by the shape of the mouse cursor, which changes from square to circular. The questions asked by the design system are supposed to be self-explanatory. Some of the items which will be added in future releases are already implemented in the menu system. Unfortunately clicking one of these options will only result in the display of the message: 'Sorry, option not yet implemented.'

3.2 Models with a Full Cut-out

This section will describe the extra input required for the generation of a 360 degree model, or a 180 degree model with a full cut-out. The starting angle of the fuselage should be set equal to 0 degrees and the ending angle to 360 degree for a 360 degree model. For an 180 degree model with a full cut-out the starting angle of the fuselage should be set equal to 270 (or minus 90) degree and the ending angle to 90 degree. These items are described in detail in appendix C. The number of parameters depends on whether the fuselage is a single circular fuselage or a double-bubble fuselage. Double-bubble fuselages are only available for models with a full height cut-out.

When a full height cut-out model is generated two extra options are offered. These are the generation of a floor structure and the generation of a double-bubble fuselage (figure 10). The floor cross-beam is modelled as a beam structure within the frame, figure 11. The

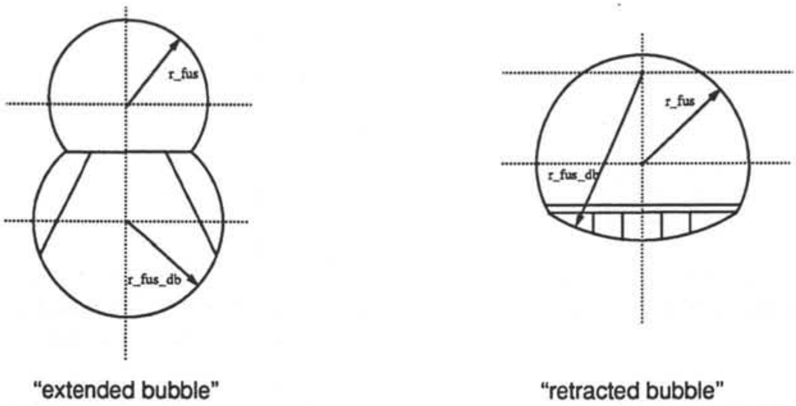


Figure 10: Double-bubble fuselage

floor itself is modelled as a built-up beam, i.e. using two beams connected by a web. This implies that the floor structure itself is strongly simplified. The main goal of this simplified finite element representation is to add a correct stiffness to the frames. The stress calculation of

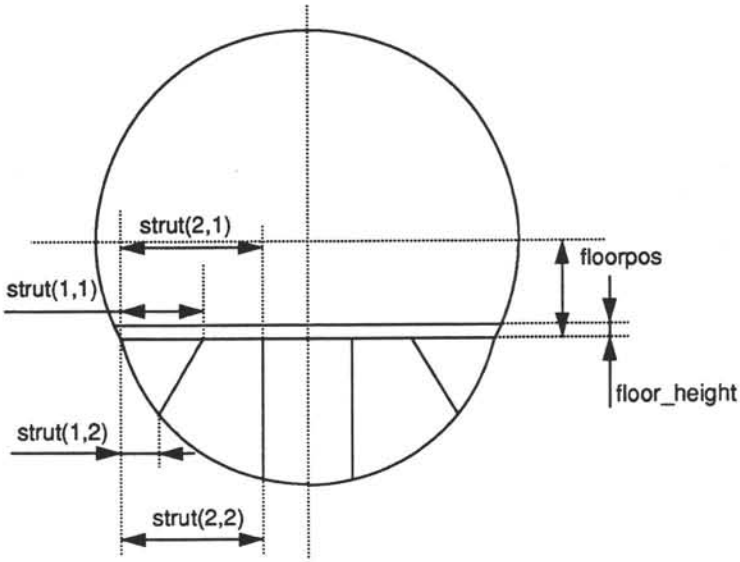


Figure 11: Definition of the floor variables

the floor is considered to be of less importance here. The variables indicated in the figures 10 and 11 are described in detail in appendix C.

The struts can also be used for full-height or crash-beams. Full-height beams are beams lying in the axial direction of the fuselage, occupying the complete height from floor to the under side of the fuselage shell. This type of beam is sometimes used in fuselages where the space under the floor is too small to store luggage. The full-height beams are built-up as two beam elements connected by a web. The mesh adaption taking place on the fuselage shell also influences the full-height beams, as well as the floor beam, which use also a built-up beam model. Naturally the mesh adaption can also take place on the elements of the floor structure, which may then result in an adaption of the mesh in the fuselage shell.

Connection of the two parts of the fuselage of different radii usually takes place at the floor position. However, the user is not obliged to place the floor at the intersection points of the two fuselage parts. It is also possible to generate a double-bubble fuselage without a floor-structure.

3.3 Cut-out Shapes

Besides rectangular cut-outs with rounded corners, other cut-out shapes can be used. There are two standard options offered. The first is the already mentioned option for rectangular cut-outs with rounded corners, while the second one generates an elliptical (or circular) cut-out. This last option is intended mainly for generating windows.

Besides the two standard options a non-standard option is offered. This option is the generation of an user-defined cut-out shape. This option will be implemented in future releases.

It is obvious that different cut-out shapes require different input. The input for the several shapes is described in detail in appendix C.

3.4 Geometry Editing

Geometry editing is the only way to change the initially created geometry. The submenu can be accessed directly from the Cufus root menu, figure 9. At this moment this menu offers three options to modify the geometry of the finite element model. Two options are used to add an offset to structural members and the last is a major change in geometry of the cut-out.

Adding an offset to the frames is meant to be an easy way to create frames with a separate castellation. The frame is modelled with a shell element for the castellation and a beam element for the frame itself. This type of finite element representation is meant primarily for the modelling of floating frames. The behaviour of a floating frame can be described much better using this representation than when it is modelled as an eccentric beam element [9].

The user has only to click the frame on which the offset should be applied, and then enter the magnitude of the offset. At the moment of writing this manual the function is not yet completely functional.

Adding an offset to the stringers should be done in the same way as above, but then for the stringers. This feature is not yet implemented. It is assumed that changing the offset of the stringers with a kind a castellation has only little effect on the stress distribution, especially in comparison with adding an offset to the frames. So this extension has a low priority.

The last option of geometry editing, the possibility to regenerate a part of the cut-out structure, will be one of the most used geometry editing features. This is partly due to the fact that the radius of the corner of the cut-out is in particular a parameter with a large influence on the stress distribution. The following parameters can be changed: width of the cut-out, height of the cut-out, the position of the cut-out and the radius of the corner of the cut-out. All these parameters can be changed within certain limits. This is due to the method of redefining the finite element model. Only a part of the finite element model is changed, the frames and stringers must remain at the same position. Because the positions of the frames and stringers are fixed, the changes in the cut-out layout are limited. The upper and lower bounds on the change in cut-out dimensions are given by the program. This partial redefinition of the finite element model is in fact a partial regeneration of the finite element model.

The earlier made mesh refinements will be saved, and will also be applied on the redefined patches. It should be mentioned that the mesh refinements are linked to the patches and *not* to the actual element size. This means that when the dimensions of a patch are changed, it is possible that a few additional mesh adaptions are required. A (partial) regeneration of the finite element model also requires a new "node equivalence", see section 4.3.1.

A disadvantage of the geometry editing of the cut-out dimensions is that the property data cannot be saved. The information for an eventual MSC/NASTRAN optimization or sensitivity analysis is also (partly) lost. This is caused by the fact that the property identification number is used for the definition of some data for the optimization and sensitivity analysis.

The partial loss of data is the result of the fact that the element numbering is not same for the regenerated finite element model and the preceding model. The element data are linked to element numbers, while it would be desirable to have a property definition coupled to the position of an element. The mesh adaption is coupled to the patches and the numbering of the patches is controlled by Cufus, so that this data will not be lost.

4 Interactive Modelling

4.1 Introduction

As stated in section 2.3 there are three different tasks of the program to be distinguished: 1) initial modelling, 2)model editing and definition, 3) post-processing. The second part, the interactive model editing and definition is almost completely controlled by the menu system. In this chapter both the interactive modelling and the interactive model definition are dealt with.

The interactive model *editing* consists mainly of making interactive modifications to the model generated by the initial model generation, and to add certain properties which were not yet defined during the initial modelling. The model *definition* consists of defining (extra) properties which are needed to perform a sensitivity analysis or a structural optimization. Although the data preparation for a MSC/NASTRAN analysis (i.e. linear static, non-linear static, sensitivity and optimization) overlaps both subjects, it will be treated with the model definition.

4.2 Element Editing

In this section the mesh adaption is discussed. This part of the program is a very powerful tool to adjust the finite element mesh. The mesh can be refined or coarsened on a user indicated patch or line[1]. Adjustment within the patches can take place in the axial direction, circumferential direction or both directions simultaneously. Changes can be undone any time by using the menu choice opposite to the previous action. For example when a patch is refined in both directions by mistake, the option "coarse mesh both" can be picked for the patch in question to correct the mistake.

1. A patch is a geometric surface definition in PATRAN. By a line is meant an one dimensional curve in PATRAN [7].

Almost all data needed is available through the standard PATRAN data base. Unfortunately, for each interactive mesh refinement an almost complete search through the data base is required. To avoid unnecessary irritation to the user, a complete search through the database during the initial generation of the model is chosen. The data found will be stored in the core memory of the computer and will be available much faster than before. This data consists of information on which patches surround each patch, and which lines are on the boundaries of each patch. Furthermore the number of elements on each patch in both directions should be available. The position of power zooms (transition regions) for each patch should be known. Finally the side numbering must be known.The generation of data concerning the adjacent patches and lines takes place at the end of the initial model generation. This process is visualized using colours for patches and lines. Storage of this data in core is only necessary once per model. The data will be written to the Cufus data base, and are therefore available for the next session.

During mesh adaption the connectivity between the patches and the lines is guaranteed. This implies that transition regions for the patches are required. Information concerning the mesh densities of the patches and the lines is stored in the Cufus internal data base (see also sections 2.5 and 6.1). If information in this data base is no longer valid, for example a wrong Cufus data base is loaded, mesh adaption can no longer take place. However, if the finite element model has the right mesh density and does not have to be changed any more, the design session can be continued.

4.2.1 Element Editing: Editing Shells

During the mesh adjustment within the patch the necessary changes in the patch itself and/or the surrounding patches, to assure a proper connectivity, are automatically generated. That means that transition meshes are generated on the patch itself or on the surrounding patches. The generation of transition patches will only use QUAD-elements, because triangular shell elements are known to produce a slightly higher stiffness than quadrilateral shell elements. The program prevents the use of mesh adjustment when triangular elements are required for the transition meshes. When further adjustments are made the program tries to remove earlier generated transition regions wherever possible, in order to obtain a mesh which is as regular as possible.

The frames, which are positioned on lines on the boundaries of most patches, are automatically regenerated after a mesh adjustment of one of the patches. The stringers are not defined on separate lines, but are generated on the patches, just like the shell elements. Because of that, no special care is required to assure the connectivity of the stringers.

The shell editing menu can be accessed directly from the Cufus main menu, figure 12.

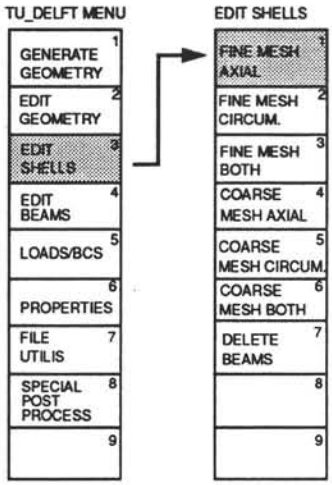


Figure 12: Interactive mesh adaption (shell elements)

The seventh option in the menu is used to delete obsolete beam elements. By obsolete beam elements is meant the beam elements lying in the axial direction of the fuselage, and which are not positioned at actual stringer locations. These elements are generated during mesh adaption. To eliminate the use of lines, which are geometric entities within PATRAN, there are beams generated on each patch in the axial direction. Because only the beams at the sides of the patch are really stringers, the beams at the centre of the patch should be deleted. This functions reads the data base for all beam elements, and is therefore too slow for use after each interactive mesh adjustment. The user has to click the option "delete beams" when he feels that the mesh adjustment is completed. However, it has no consequence if the beam deletion is performed more than once.

Two patches with a common edge will both generate a stringer along that common edge. Therefore, after a mesh adaption, deleting obsolete beams and node equivalencing, some duplicate beam elements will exist at the sides of some adapted patches. These elements can be automatically removed by a PATRAN function, accessed as shown in figure 13. It must be emphasized that these duplicate elements can only be found if node equivalencing has taken place.

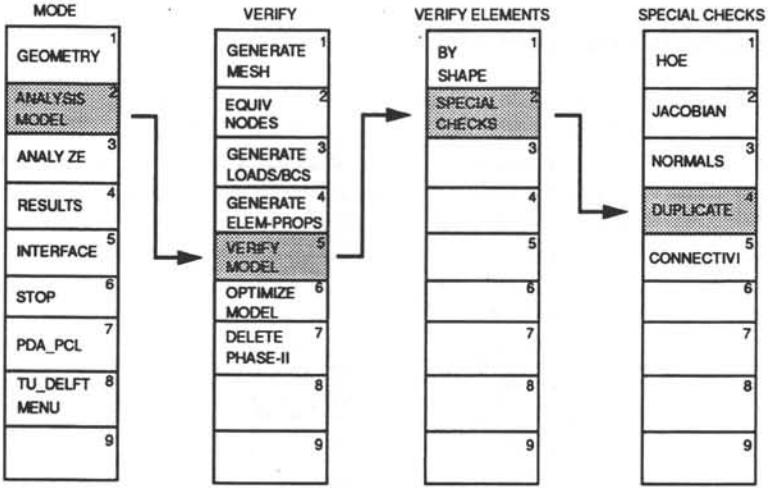


Figure 13: Removing duplicate elements after mesh adaption, removal of obsolete beams and node equivalencing. (Standard PATRAN menu)

There are some limitations on the use of the mesh adjustment. First of all mesh adjustments are only capable of doubling the number of elements in a certain direction or halving the number of elements in a certain direction. This means that a patch with an uneven number of elements in a certain direction cannot be coarsened. Because the mesh refinement only deals with doubling the number of elements in certain directions, a rapid increase of number of elements can occur. Some of the flexibility in modelling transition regions is sacrificed to simplicity of programming and simplicity of running the program. If the user wishes to create another type of mesh adaption than Cufus offers, it can be performed in the "standard PATRAN way". However, after a "manually" performed mesh modification, a mesh adaption performed by Cufus will be impossible. All other features within Cufus are not affected by a "manually" performed mesh modification.

4.2.2 Element Editing: Editing Beams

Beam editing consists of an adjustment of a line with beam elements. This line can be coarsened or refined. It is intended as an extension of the shell editing described in the previous section. Most of the necessary adjustments are performed when the patches are modified with shell editing. Because separate modelling adjustments of the beams are regarded as less important, this item is not yet implemented. The menu options are already implemented, figure 14.

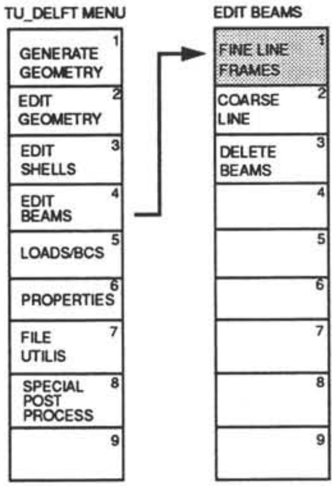


Figure 14: Interactive mesh adaption (beam elements).

4.3 Loads and Boundary Conditions

Menu option 5 from the Cufus main menu offers the user an easy application of loads and boundary conditions for a number of predefined load cases. Furthermore there are some additional tools provided by this option. These options are the generation of extensions to the MSC/NASTRAN Bulk Data Deck and the generation of a MSC/NASTRAN Case Control Deck. There is also an option called "node equivalence", which connects the separately meshed geometric entities, like patches and lines.

4.3.1 Loads and Boundary Conditions: Node Equivalencing

Node equivalencing means that nodes which have the same position in space are merged into one node. The same position means in this case that the positions must be the same within a certain user-defined tolerance. Normally PATRAN generates meshes on to geometric entities. This implies that meshes on patches and lines are mutual disconnected. However, the nodes on the edge of patches do have the same coordinates as its adjacent patch or line. The process of node equivalencing joins the different parts of the mesh to a coherent finite element model.

Node equivalencing cannot be accessed through a PCL function, only by using the standard PATRAN menu. Cufus itself 'travels' to the PATRAN menu for node equivalencing, executes the node equivalencing and travels back to the Cufus menu from which it was launched. If node equivalencing is started, a tolerance is proposed in the alpha area. A reasonable value for the tolerance for normal fuselages is 0.2 mm. This value can be changed by setting the PATRAN variable TOL to another value [8]. However, attention should be paid to the fact that this tolerance parameter is not only used for node equivalencing but also for geometry generation.

The menu appears as shown in figure 15. After node equivalencing the user is asked if the changes due to the node equivalencing should be reflected in the active set. If the answer to that question is "no", equivalencing does not affect the finite element model. In normal use the answer to that question should be "yes".

It does not matter if a finite element model is equivalenced more than once. Mesh adaption can take place both before and after node equivalencing. Care should be taken when loads and/or boundary conditions are applied. It is possible that loads and boundary conditions are connected to nodes which after equivalencing are removed from the model. Therefore it is a useful habit to apply loads and boundary conditions ***after node equivalencing.*** The menu has such a lay-out that the processes are placed in a logical top to bottom order.

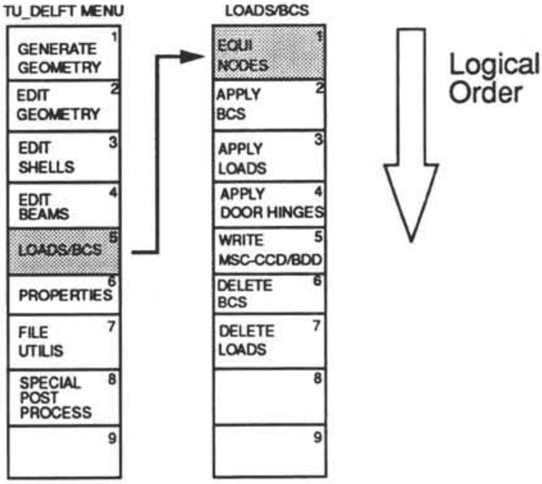


Figure 15: Node equivalencing.

4.3.2 Applying Loads and Boundary Conditions

Boundary conditions can be applied interactively for a limited number of standard load cases. The set-number for the boundary conditions is used for identification of the type boundary conditions and is therefore fixed. The program uses these fixed set-numbers, when generating a MSC/NASTRAN Case Control Deck. Load cases also have a fixed set-identification number so that for each load case automatically the right set of boundary conditions in combination with the corresponding set of loads is chosen. By set of boundary conditions is meant a set of SPC's in combination with a set of MPC's. SPC is the abbreviation of single point constraint, which are applied on the (anti-) symmetry edges of the structure. MPC's (multi-point constraints) are used for an proper load introduction at the end of the shell. The following boundary conditions can be applied interactively:

1) Internal pressure & uniform axial load

2) Transverse force along the y-axis & bending moment about the x-axis

3) Transverse force along the x-axis & bending moment about the y-axis

4) Torsional moment.

The loads can only be applied on the edges of the finite element model. If the user wishes to apply also forces on for example the frames, this should be performed by hand. By giving the loads the same identification numbers as the pre-defined Cufus identification numbers, these manual added loads will be included automatically in the Cufus load set.

Finally the menu offers the option to apply all types of boundary conditions at once. This option also applies the boundary conditions with fixed set-numbers. How to access the menu to apply the boundary conditions is shown in figure 16. This figure also indicates that there are menu options for the deletion of all boundary conditions and loads.

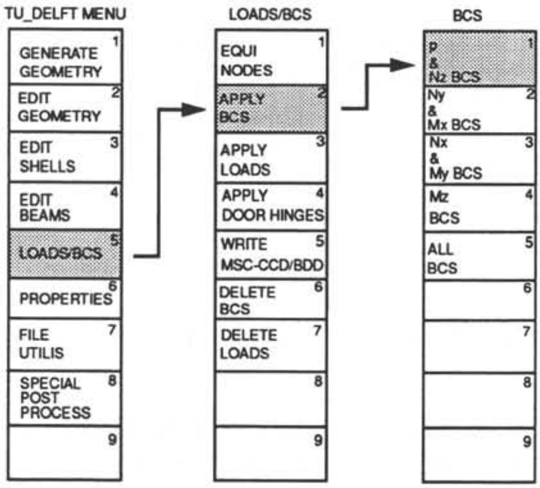


Figure 16: Interactive application of boundary conditions.

Loads and boundary conditions can be applied in a number of different ways. Loads can be applied by using the menu to define the loads interactively, see figure 17. The application can be performed by using an input file, see also figure 17, and of course by a combination of the first two mentioned options for load application. Some information about the format and contents of the input files can be found in and appendix C. Furthermore, it may be clear that the sets of boundary conditions may be applied or changed by using standard PATRAN commands. It is recommended in any case to use the fixed set-identification numbers because in that case Cufus links the sets of loads with the matching boundary conditions. This is because during the creation of a MSC/NASTRAN Case Control Deck these identification numbers are used, whether they are present or not. The fixed set-identification numbers

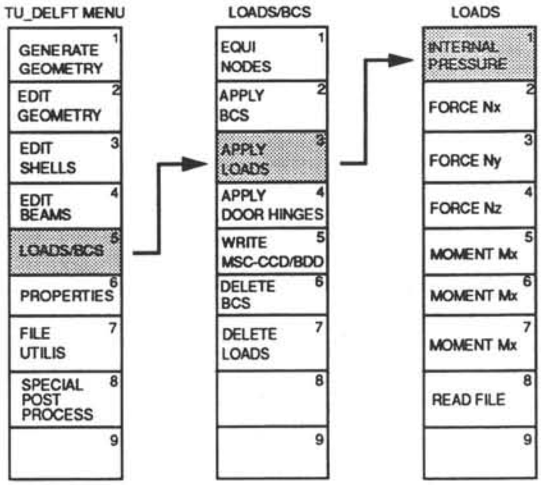


Figure 17: Interactive application of loads.

are given in table 1. Loads can also be defined by standard PATRAN commands, but in that case the user must write the Case Control Deck, because it cannot be generated by Cufus. The sign conventions are shown in figure 18.

Type of Load	Boundary condition set identification number
Internal pressure and axial load (p & N_x)	11
Shear load and bending moment (N_y & M_x)	21
Shear load and bending moment (N_x & M_y)	31
Torsional moment (M_z)	41

Table 1: Identification numbers loads and boundary conditions.

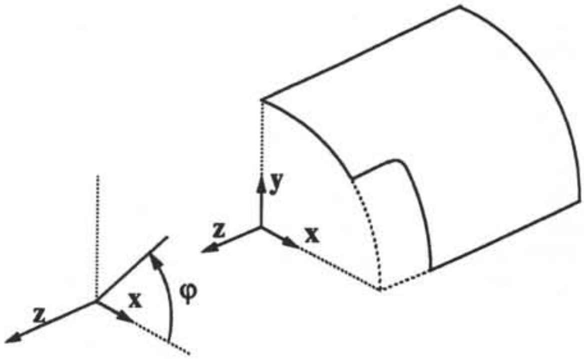


Figure 18: Sign conventions.

4.3.3 Loads and Boundary conditions: MSC/NASTRAN Case Control Deck and Bulk Data Deck

PATRAN is able to generate a MSC/NASTRAN Bulk Data Deck. This Bulk Data Deck contains all the information about the finite element model itself. The interface PATNAS [10] generates the Bulk Data Deck by translating the neutral model file. This neutral file is standard output generated by PATRAN and is meant to be fully transparent for a number of finite element packages. Unfortunately it is (still) no common practice that each finite element program has an integrated optimization and/or sensitivity analysis like MSC/NASTRAN. That is no doubt one of the reasons that PATRAN cannot define design variables, constraints and objective functions.

Cufus, however, is able to define all data necessary for an optimization or sensitivity analysis. These data could perhaps be included in the neutral model file, but cannot be recognized by the PATNAS interface. Therefore a part of Cufus also generates an additional part of the MSC/NASTRAN Bulk Data Deck. This extra part of Bulk Data Deck can be inserted in the Bulk Data Deck generated by PATNAS, by using a text editor or a small extra program. To maintain a maximum of user comfort, it would be preferable to have a program or shell-script which contains all the necessary commands to execute the different programs to connect

PATRAN with NASTRAN. The extension to the Bulk Data Deck and the Case Control Deck can be accessed by the menu as shown in figure 19.

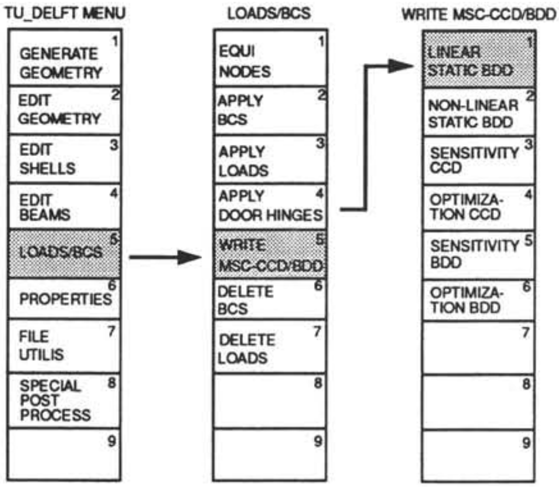


Figure 19: Generation of the MSC/NASTRAN Case Control Deck and the extension to the Bulk Data Deck.

When the extension of the Bulk Data Deck is generated for a sensitivity analysis, some extra internal data is also generated. For a sensitivity analysis the constraints are intended to apply to the complete model. MSC/NASTRAN defines the constraints for a sensitivity analysis for each element. The user only has to apply the sensitivity constraint to a single element. This extra generation is performed when writing the extension to the Bulk Data Deck. Some extra information needed for the post-processing of the sensitivity results is also generated during this procedure, and is stored in an array. This array is also written to the restart file. When this information is lost, it can be recovered by generating the extension to the MSC/NASTRAN Bulk Data Deck again.

The Case Control Deck contains information to control the NASTRAN job. In this part of the input file the solution sequence (i.e. the type of analysis) is chosen. Furthermore this is where to select the set of boundary conditions and loads is selected. This part of the input file can be generated by Cufus. It selects the right set of boundary conditions, with the corresponding set-identifications of the loads. It also adds the solution sequence. The standard chosen solution sequences are now SOL 101 for geometrically linear static and sensitivity analysis, SOL 106 for a geometrically non-linear static analysis and SOL 200 for an optimization.

The name "Case Control Deck generated by Cufus" is strictly speaking not correct, because Cufus generates not only the Case Control Deck but also an Executive Control Deck. Because this part is quite small it is for convenience included in the Case Control Deck.

Some data in the Case Control Deck cannot be determined exactly. For example the maximum CPU-time should be estimated (this is the "time"-card on the Executive Control Deck, preceding the Case Control Deck). This is highly dependent on the type of computer which runs NASTRAN. Furthermore the size of model strongly influences the needed CPU-time. The value set by Cufus is a value based on experience running with a Sparc 1000 multi-processor computer. The time set by Cufus is therefore no more than a recommendation. Some output requests are also made by Cufus. These requests will in general be sufficient, but some individual changes are always possible.

Some of the values of the following parameters are also chosen on basis of experience. These parameters sometimes may need to be changed. The parameters referred to are some optimization parameters mentioned on the DOPTRM-card [11]. The parameters which are set by Cufus are the maximum number of iterations and the finite difference step for the determination of the sensitivity data. When an optimization is stopped because the maximum number of iterations is reached, this is clearly indicated in the user output generated by NASTRAN. When an optimization stops before it reaches an optimum, it can sometimes be corrected by increasing the value of the finite difference step. This is caused by the way the sensitivity data is computed. In order to calculate the sensitivity data the following equation must be solved.

$$[K]\frac{\partial \underline{U}}{\partial x_i} = \frac{\partial \underline{P}}{\partial x_i} - \frac{\partial}{\partial x_i}[K]\,\underline{U} \tag{1}$$

In equation (1) [K] is the stiffness matrix, x_i is the design variable i, $\underline{U}$ is the displacement vector, and $\underline{P}$ is the is the load vector. The derivatives $\frac{\partial \underline{P}}{\partial x_i}$ and $\frac{\partial}{\partial x_i}[K]$ must be determined before equation (1) can be solved. These derivatives are approximated by first forward differences [12].

$$\frac{\partial}{\partial x_i}[K] \approx \frac{[K]\ (\underline{x}+\Delta x_i) - [K]\ (\underline{x})}{\Delta x_i} \tag{2}$$

$$\frac{\partial \underline{P}}{\partial x_i} \approx \frac{P\,(\underline{x}+\Delta x_i) - P\,(\underline{x})}{\Delta x_i} \tag{3}$$

Solving the formulae (2) and (3) results in a pseudo-load vector which can be substituted in equation (1). The calculation of the finite differences can easily result in an underflow if the finite difference step is chosen too small. The default settings for the finite difference step (Δx_i equals 0.01 x_i, DOPTPRM-card [11]) may result in a prematurely terminated optimization. Increasing the finite difference step to 0.05 times x_i is normally satisfactory. It must be noted that increasing the finite difference step also increases the truncation error, which is $O(\Delta x_i)$. It can be stated that an optimum finite difference step must exist. Most optimizations are not extremely sensitive to the value of the finite difference step, but in some (rare) cases the effect can be significant, as shown in figure 20.

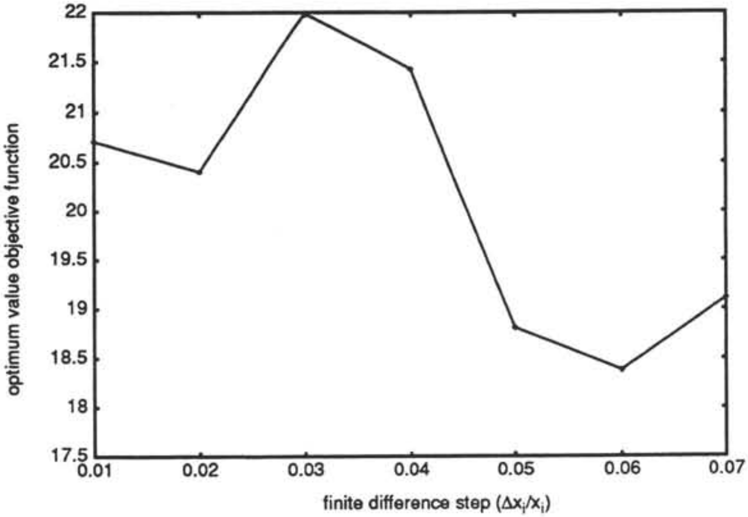


Figure 20: Effect of the finite difference step for an optimization.

Finally a few remarks are sometimes made in the Case Control Deck file produced by Cufus. These remarks normally contain a few comment lines which can be inserted in the Bulk Data Deck. Because this replacement is only a very few lines, this method of manual editing is considered easier than writing it to a separate file and joining it with the Bulk Data Deck produced by PATNAS. Obviously, to effect these changes the comment signs in the lines copied to the Bulk Data Deck must be removed.

4.4 Property Definition

A large effort is made for a user-friendly interface for entering property data for the finite element model. Globally the property definition can be divided in two parts, first the definition of the properties themselves and secondly the assignment of property data to certain elements. Property definition means that certain properties, like for example the thickness, is coupled to a property identification number. The property assignment refers to the linking of property identification numbers to one or more elements.

Especially the assignment of properties to certain elements is very user-friendly by taking full advantage of the menu capabilities and mouse controlling capabilities. The definition of the properties is only partly user-friendly. For a number of standard beam cross-sections the properties can be calculated. These properties are calculated in a parametric format, so that besides standard dimensions, arbitrary dimensions are also possible.

The other way of entering property definitions is by reading a user defined file. This file contains the property data in a format which is in fact just as user-unfriendly as the original PATRAN-format. The idea behind a file with all property data is to create a kind of data base which contains, for instance, all used types of stringers, or all commercially available skin thicknesses. Besides property data, also material data can be entered in this file. It is not mandatory to use this file for the material and element property definition, the user is always free to use the conventional way in PATRAN to define material properties and element properties.

Property definition is dependent on the analysis code used. Therefore it must be emphasized that the property definition in Cufus (i.e. standard beam cross-sections up to now) will only work properly in combination with MSC/NASTRAN.

The property entries are also "abused" for the definition of the design variables. Because PATRAN does not recognize design variables, property entries with a reserved property identification number are used for the definition of the design variables. For this reason the section concerning the definition of the design variables and the constraints is also included in this chapter, see section 4.4.3.

4.4.1 Property Definition: Using Predefined Properties

As stated in the previous section, property data can be defined in several ways. The way of defining property data in the standard PATRAN form will not be discussed here because it is covered in the PATRAN User's Manual [9].

Another form of property definition is to create a file with pre-defined properties. The investment of some time in creating a data base of materials and properties can sometimes be very useful. Because the definition of property data is not less user-unfriendly than the standard PATRAN way of property definition, it will only be effective by repeated use of Cufus. However, if in future more design tools similar to Cufus are created, it is very likely that the format of property files can remain unchanged. In that situation it will be very profitable to have a predefined property data base. Anyway, only in the worst cases will it take as much time using the property definition file as using the standard PATRAN way of defining properties. In all other cases using the property definition file will result in a slight to even considerable gain of time. The format and contents of the property input file is given in appendix C.

For a number of standard beam cross-sections a complete property definition is available in parametric format. Parametric format means that the cross-sectional property data are available as functions of the dimensions of the cross-section of the beam. This implies that besides the standard dimensions also beam sections with arbitrary dimensions can be chosen. The definition of these beams works with a menu system which is partly written in the graphical window of PATRAN. This part of the menu is accessed as is indicated in figure 21. Clicking the option indicated in this figure will cause the display of a picture over the finite element window. In this picture a number of different beam cross-sections are drawn as given in figure 22. At this moment only four different beam types are included. These beam types [13] with a short description are listed in table 2. In future releases this number will be increased. Simply by clicking one of the

Beam number	Description
FoN 1.6302	Formed plate channel section
FoN 1.6304	Formed plate Z-section
FoN 1.6307	Formed plate hat section.
FoN 1.6308	Formed plate Z-section

Table 2: The currently implemented standard beam sections.

squares in which the beams are drawn, the next graphics menu will appear. This menu consists of a picture with the typical dimensions and a list of standard dimensions. In the list with standard

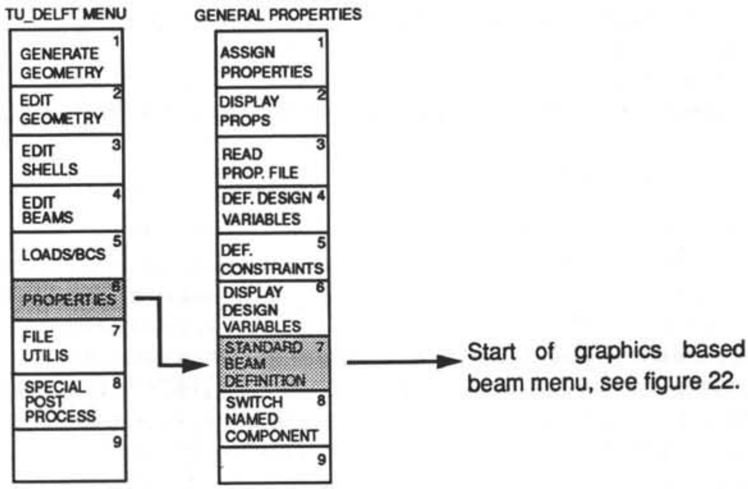


Figure 21: Start the standard beam property definition.

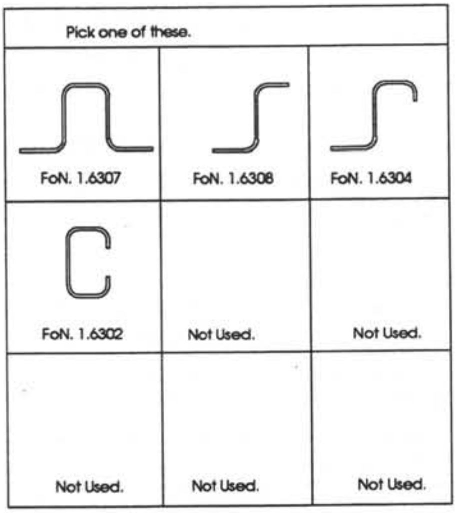
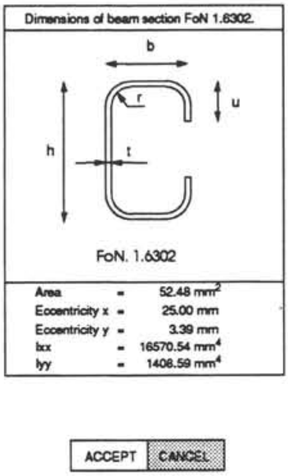


Figure 22: The starting menu for the definition of standard beam properties

dimensions an option called "User defined" is included. This option offers the opportunity to define a non-standard beam cross-section by entering values for the dimensions indicated in the picture with the typical dimensions. As example beam cross-section FoN 1.6302 is chosen, see figure 23. A standard beam cross-section can be chosen by simply clicking the table cell. In that



h	b	u	t	r
25.	12.	4.0	0.6	1.6
25.	15.	6.0	0.8	2.5
25.	15.	6.0	1.0	2.5
25.	15.	6.0	0.6	1.6
25.	15.	6.0	0.8	2.5
25.	15.	6.0	1.0	2.5
25.	15.	6.0	1.2	4.0
25.	15.	6.0	1.6	4.0
30.	12.	4.0	0.6	1.6
30.	15.	6.0	0.8	2.5
30.	20.	8.0	1.2	4.0
30.	20.	8.0	1.6	4.0
40.	15.	6.0	0.8	2.5
40.	20.	8.0	1.2	4.0
40.	20.	8.0	1.6	4.0
50.	15.	6.0	0.8	2.5
50.	15.	6.0	1.0	2.5
50.	20.	8.0	1.2	4.0
50.	20.	8.0	1.6	4.0
More options.				

Figure 23: One of the sub-menu's for the definition of standard beam properties

case the table cell will be drawn in red and in the extreme left box some typical properties (like area, eccentricities and second moments of inertia) are displayed. This allows the engineer to define a standard beam which better matches the requirements.

A small box with the options "ACCEPT" and "CANCEL" is placed below the left box. This option is used to accept a certain cross-section. If "CANCEL" is chosen another cross-section shown can be picked from the standard list. This allows the user to play around with different cross-sections in a very easy way. If the option "ACCEPT" is picked the property definition will continue. The identification number for the material used and the property must be supplied. If a property identification number already exists, the user has to confirm explicitly that he wishes to overwrite the old property entry. If that confirmation is denied, the property definition will be aborted. The material to which the property will refer does not have to be defined yet, but must be defined before the finite element model is translated to MSC/NASTRAN, of course.

In figure 23 an option called "More options" is included in the dimension list. This option is used for dimension lists which were too large to be displayed in one list. Clicking this option will display the rest of the dimension list. Clicking the option "More options" again the first part of the list will appear again.

The graphical menu is built up using the graphics function library in PCL. This causes the menu to be drawn over the finite element model. However, the display of the menu does not affect the finite element model. The model can be drawn again by clicking the plot option in the menu at the top of the PATRAN window.

4.4.2 Property Definition: Assignment

Normally property definition is followed by property assignment. However, the user is not restricted to this sequence. It is possible to assign the properties before the definition of these properties. However, if the user wants to assign the properties before the property definition, only standard PATRAN commands can be used. Using Cufus for property assignment does not allow to assign a non-existing property identification label to the finite element model. The main reason for this refusal is to prevent errors during the finite element calculation. The user must keep track of the property and material identification numbers used. Because the elements can be indicated graphically, contrary to properties and materials, element numbers do not have to be used by the designer.

Normally the design variable for shell elements is the thickness of the element, while the design variable for beam elements may be cross-sectional area, second moment of area about the x-axis or y-axis, or torsional stiffness. Cufus is up till now not capable of defining linked design variables. Linked design variables means, for example, that the second moment of area is a user-defined function of the cross-sectional area. Since MSC/NASTRAN is capable of dealing with such design variables it is intended to remove this limitation in future releases.

When the user wishes to assign a property to an element or a group of elements, Cufus starts by asking him to specify the property identification number. It will be recognized whether the property record connected to that number is a shell or beam property. So it will not be possible to assign, for example, beam properties to a shell element. If the user has to indicate certain entities (e.g. patches, lines or elements), only the entities for which the property record is valid are "clickable". The material identification number used for an element is included in the property record.

There are a large number of ways to assign a property to one of more elements. Table 3 lists the currently available possibilities of property assignment.

	single	group	active set	all elements
QUAD	yes	yes	yes	yes
TRI	yes	yes	yes	yes
BAR	yes	yes	yes	yes
line	yes	yes	yes	yes
patch	yes	yes	yes	yes
stringer	yes	no	no	yes
frame	yes	no	no	yes

Table 3: Property assignment possibilities.

All property assignment options are interactive and can be invoked by the menu. Figure 24 shows how, for example, a property can be assigned to all patches.

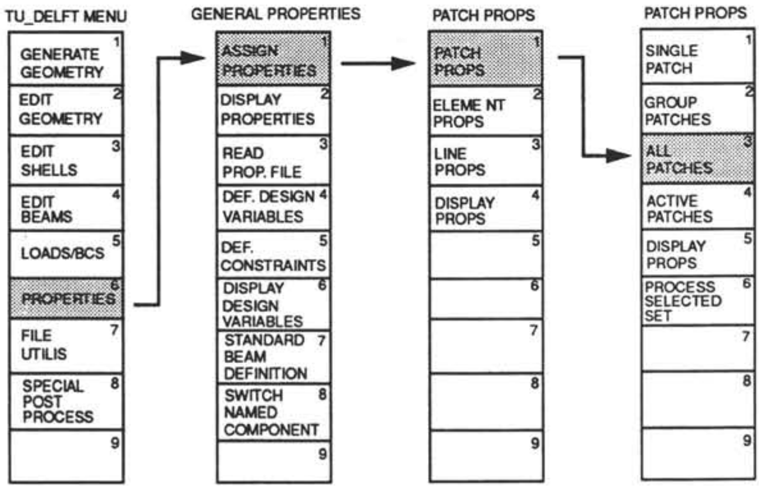


Figure 24: Example property assignment for all patches.

In the "PATCH PROPS" menu in this figure an option "PROCESS SELECTED SET" is visible. This option needs to be used when a group of patches (or lines and elements) is chosen for property assignment. When a group of entities is indicated clicking this option it will effect the changes indicated. If entities are clicked for such a group assignment, they will change colour. If an entity is clicked by mistake, it can be removed from the selected set by clicking it again. Once the selection is made it can be concluded by the previously mentioned option "process selected set". A selected set can hold up to 200 entities, which is normally more than sufficient. If a selected set needs to contain more than 200 entities, the property assignment should be split into two or more separate group assignments.

Globally this way of property assignment can be performed for patches as well as for elements and lines. A powerful option is to generate a property record for ***all*** elements, patches or lines, especially in the initial phase of the property assignment.

Property assignment for the frames can be performed in several ways. First it is possible to do an element or a line group assignment. Secondly it is possible to assign properties to all frames or to a single frame, see figure 25. The user is asked to indicate a reference ele-

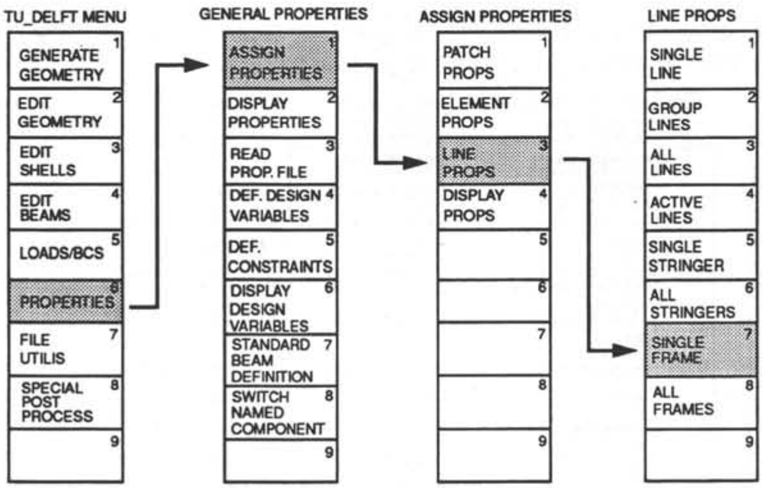


Figure 25: Assigning properties to a single frame.

ment of the frame to be changed. Cufus starts a search in the PATRAN data base to find beam[1] elements lying in the circumferential direction and positioned on the same frame position as the reference element. If an element is found for which all conditions are met, it is changed. When the conditions are tested, certain tolerances on the coordinate positions are applied. Up to now a tolerance of 5 mm is used, which for "normally" dimensioned fuselages

1. Note: The terms *BAR* and *BEAM* are mutual interchangeable in this document. This is due to the fact that these terms are interchangeable in PATRAN. In MSC/NASTRAN they both refer to one dimensional elements with extensional, torsional and bending stiffness, though there are some differences between these two elements.

will yield satisfying results. This tolerance can only be changed in one of the PCL functions [8].

The same type of property assignment is also applicable to stringers. This is especially very useful because the stringers are, contrary to the frames, not generated on a line but on the edges of the patches. This implies that properties of stringers can only be assigned by element group assignment or by the special stringer property assignment. This works similarly to the property assignment for the frames, with the exception that the data base is now searched for beam elements lying in the axial direction and positioned at the same position as the reference element.

For the frame property assignment, as well as for the stringer property assignment it is not necessary to indicate a reference element if the option "ALL FRAMES" or "ALL STRINGERS" is chosen (figure 25). In these cases the data base is scanned for beam elements in the axial direction lying on stringer positions (option "ALL STRINGERS") or for beam elements oriented in the circumferential direction lying on frame positions (option "ALL FRAMES").

For the types of property assignment for which the complete PATRAN data base should be scanned, it should be noted that these operations will require considerably more time than the other types of property assignment, depending on the size of the finite element model.

The user has to make a table (or something similar) of property identification numbers corresponding of property definitions, and a table with property definitions with corresponding material identification numbers in the property definition file.

Properties identification numbers assigned to the finite element model can be made visible in a "fill-hide" plot. The possibility to make a plot with property identification numbers is offered in most menus concerning the property assignment. Figure 26 shows how these can be reached in the main property menu. The figure shows the several options for the display of the property identification numbers. Most element types can be displayed separately or in combination with other types of elements. Displaying separate element types can be especially useful for displaying beam properties because, in a plot of beam and shell properties combined, beam properties are normally hard to distinguish.

Plotting the properties makes use of the automatic colour assignment. This could give the same colours to elements with different property identification numbers due to the limited number of colours available. In the future an option will be added to display all property identification numbers in subsequent plots, regardless of the number of property entries. Unfortunately when different elements have the same property ***data*** but different property ***identification numbers***, they are displayed as different properties.

Finally it should be mentioned that by using output in the alpha-area from PATRAN, all questions considering assignment of property data to an element, patch or line which already has a property record can be positively answered by the program. This means that old property data

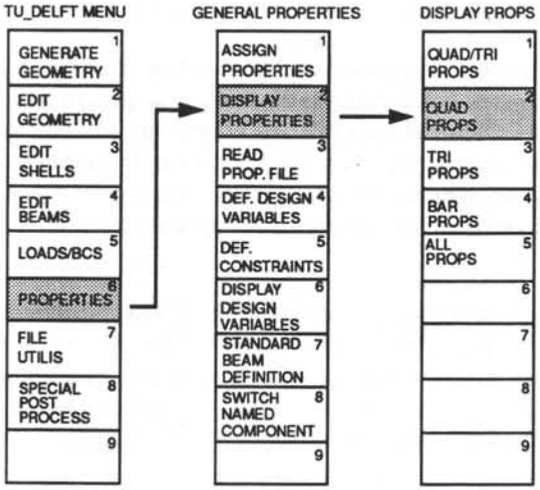


Figure 26: Displaying shell properties of the QUAD elements.

can be overwritten without any warning. This seems undesirable, but in practice has been proved to be preferable to giving an explicit confirmation before each element is changed. If a property is assigned by mistake, it is easy to correct.

4.4.3 Property Definition: Definition of Design Variables

In order to run a structural optimization or a sensitivity analysis with MSC/NASTRAN it is necessary to define design variables and constraints, and in the case of an optimization also an object function. The definition of the design variables is exactly the same for both analysis types. The definition of design variables is very similar to the property assignment. The existing property entry will be copied to a new entry with an identification number above 4000. All properties with such a identification number are assumed to be design variables. Therefore it is important that the user only defines properties with identification numbers *below* 4000, to prevent interference with the pre-processing of design sensitivity or structural optimization jobs. Although the definition of design variables is similar to that of the property assignment, the number of ways in which design variables can be defined is somewhat limited compared to the number of ways for property assignment. Figure 27 shows how the menu for the definition of the design variables can be found.

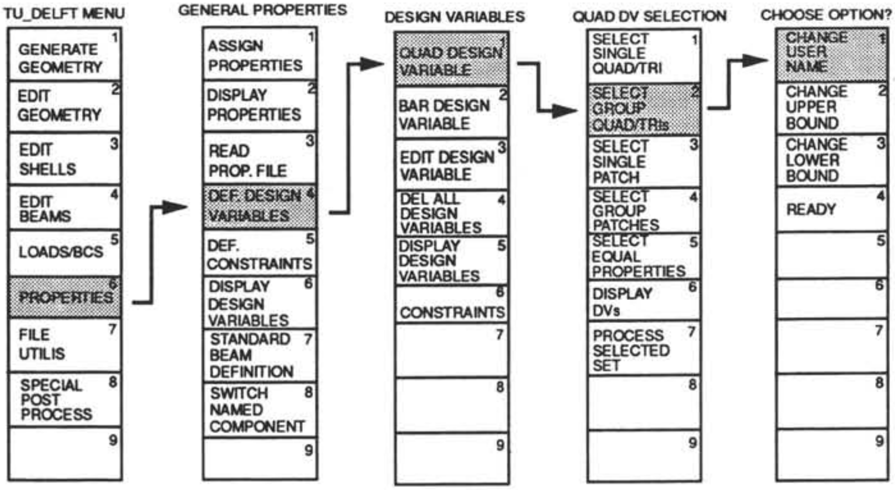


Figure 27: Definition of the user name of a design variable assigned to a number of shell elements.

When an element with a property identification above 4000 is selected for definition as a design variable, the program notifies the user by displaying a warning but continues anyway. In this manner it is possible to overwrite existing design variables.

The set up for the definition of the design variables is not as flexible as the property assignment. This is due to the fact that the numbering of design variables is automatic, unlike the numbering of the property identifications which is completely user controlled. This implies that the user does not have to be bothered by keeping track of design variable numbers, but the price to be paid for this is a somewhat less flexible way of defining design variables. An example of these limitations is the impossibility of performing a combined assignment such as a group of patches combined with a group of elements.

Once the design variables are defined, they can be modified using the menu option "EDIT DESIGN VARIABLE", see figure 27. This option allows the user to edit the name of the design variable and its upper and lower boundaries. If the design variable references to a BAR element, it is also possible to change the type of design variable, for example change the design variable from cress-sectional area to second moment of area. The design variable to be changed can only be indicated by typing its identification number. The names of the design variables along with its properties are shown in the alpha area. Identification by the design variable number is somewhat user-unfriendly, but since the number of design variables is normally limited, this will be a surmountable problem. if the number does not correspond with the design variable the user intends to change, its easy to skip this design variable number and continue with the next design variable.

The number of elements belonging to a design variable cannot be increased; if one wants to do so, then it is necessary to define the complete set of elements again. However, there is a little "trick" to make a modification to the group of elements belonging to a design variable. Since the design variables use property entries, the property assignment can be used to increase the number of elements belonging to a design variable. By displaying the design variables the property entries are plotted. From this plot the property entry of the design variable in question can be found. By assigning the property identification number to the elements which have to be part of the design variable, the modification of the design variable is completed.

If the user wants to define a complete new set of design variables, it is possible to delete all design variables, using the option "DEL ALL DESIGN VARIABLES", see figure 27. Attention has to be paid to the fact that only the information in the Cufus data base is erased. The property identification numbers of the elements belonging to the former design variables are still in the range above 4000. This implies that the property identification numbers should be changed by the user.

The menu also shows an option to display the design variables. In fact this is a fill-hide plot of the property identifications, for which only the property identifications above 4000 are shown. The numbers on the list of colours displayed on the right of the screen point to these property identification numbers. The number of the design variable is the number of the property identification number minus 4000. The assignment of colours is done automatically by Cufus. This has the advantage that for each design variable a separate colour is available. In fact Cufus uses the manual colour assignment option offered by PATRAN. Cufus chooses 15 colours for each plot. When there are more than 15 different values to show, the display is divided into successive plots, each showing 15 design variables, except for the last plot which can show less than 15 design variables. Only elements belonging to a design variable are shown, the other elements are erased from the display.

4.4.4 Property Definition: Definition of Constraints

Constraints for optimization and sensitivity analysis are defined differently in MSC/NASTRAN. This is not because the constraints for the two types of analysis differ basically, but because there are some different parameters needed for each type of analysis. It should be noted that there are two ways of calculating sensitivity coefficients. The calculation of the sensitivity data could be accomplished by defining an analysis type, e.g. linear static analysis or a linear eigenvalue analysis, defining extra data such as design variables and constraints, and the required sensitivity output. The second method involves a slightly modified optimization. When a certain parameter is set, optimization will stop after the sensitivity analysis has been completed. In the last case the same constraints can be used for a sensitivity analysis as well as for an optimization. Unfortunately post-processing of the data generated by the modified optimization is more difficult than with a separate sensitivity analysis. Therefore the separate sensitivity analysis is implemented, though for future releases use of the modified optimization will be considered too.

However, the user of Cufus is not troubled by the difference between the two analysis types. The constraint definition is the same for both analysis types. There are two items to which the user should pay some attention. First the element constraints (constraints on stress, strain and force) act on the complete model for a sensitivity analysis and act on all elements within a property group (i.e. group of elements with the same property identification number) for an optimization. Nodal constraints (displacement constraints) act on a single point or on the complete model, to be chosen by the user for both types of analysis. Secondly the upper and lower boundaries on the responses are not required for the sensitivity analysis. In future releases it is intended to have a fully consistent constraint definition.

All constraints required by MSC/NASTRAN and not explicitly defined by the user are generated automatically by Cufus. Especially for large models the number of automatically generated constraints can become quite large. This implies that the time needed for the data base scan, in order to generate the constraints, also increases.

Generation of extra constraints takes place during the generation of the extension of the Bulk Data Deck, see section 4.3.3. At that stage an array with sensitivity pointers is also generated. This array with pointers is essential for a correct post-processing of the sensitivity data. This pointer array is also written to the restart file, see section 2.5 and 6.1. When a pointer array is lost, for example by forgetting to save the local Cufus data base, it can be easily be regenerated by writing again an extension to the Bulk Data Deck. This extension of the bulk Data Deck is not necessary, but in this way, the sensitivity pointer array is stored again in the core memory of the computer.

The generation of element constraints and nodal constraints will be discussed separately. Figure 28 indicates how the constraint definition is invoked. The definition of the element constraint will be discussed on the basis of figure 29 which, due to limited space, starts at the third menu (column) in figure 28. The second menu column called "CONSTRAINT DEFINITION CON(7)", in figure 29, shows the options to define a constraint. The identification number of the constraint to be defined is indicated in the name of this menu, in this case constraint number 7 is being defined. The first option, "Change user name" is used to add a name (not necessarily an unique name) to a constraint to make the MSC/NASTRAN output more readable. This name has no further purpose. It is mandatory in MSC/NASTRAN to have a user name on constraint entry,

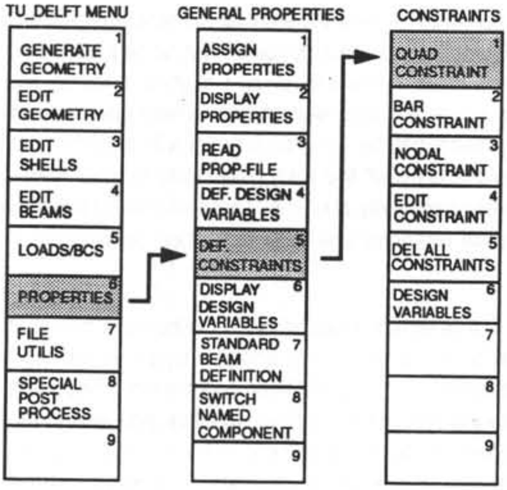


Figure 28: Start of the definition of a constraint for a quadrilateral element.

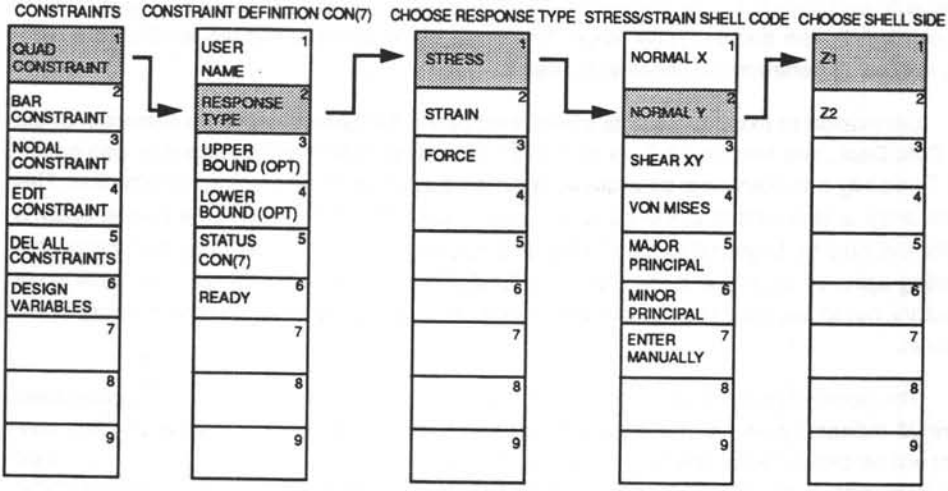


Figure 29: Definition of the response type for a constraint acting on quadrilateral shell elements.

but if the user does not enter a name, a name is supplied by Cufus. This means that MSC/NASTRAN cannot crash due to lack of a user name connected to that constraint.

The second option in the menu, "RESPONSE TYPE" will be discussed later. The third and the fourth options are used to define the upper and lower bounds for the constraint. These options only have to be used when the constraints are being used for an optimization. If these values are omitted, Cufus will give the upper and lower bounds default values, to prevent a crash of MSC/NASTRAN. Option number 5, "STATUS CON(7)", can be used to display the status of the indicated constraint, in this case constraint number 7. Displaying the status of the constraint means that all properties of the constraint will be displayed. Choosing option 6, called "READY", the definition of the current constraint will be ended.

Option number 2, "RESPONSE TYPE", is the only option which will lead to sub-menus. The response type is a description of the quantity to be constrained. In Cufus stress, strain, and force can be constrained for all elements. The options shown in the last menu column of figure 29 are self explanatory, with option "Enter manually" probably excepted. This option is meant to enter a response code which is not covered by the menu. All available response codes are listed in the MSC/NASTRAN User's Manual [11].

The definition of BEAM constraints is similar to the definition of QUAD constraints with the only exception that the response codes differ, see figure 30. The response codes in the left-most

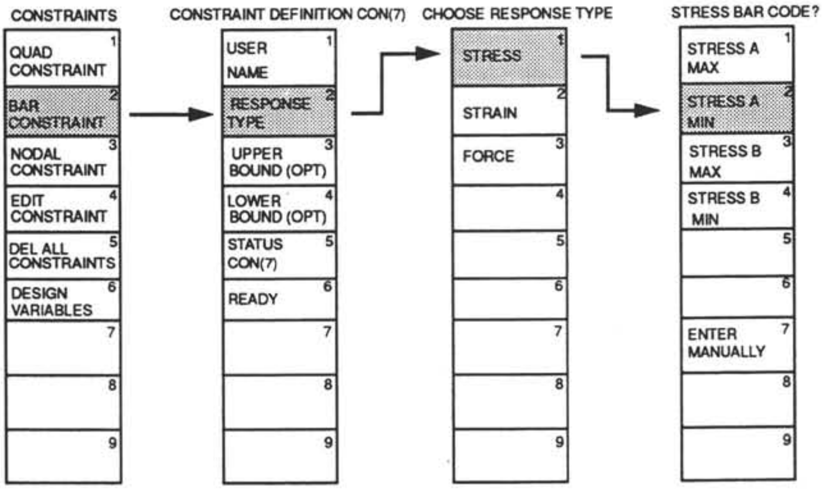


Figure 30: Definition of the response type for a constraint acting on beam elements.

menu column of this figure refer to stresses at both ends of the beam elements. In order to calculate stresses in beam elements, stress points should be defined in the PBAR- or PBEAM-card in the MSC/NASTRAN Bulk Data Deck [11]. These values are not (yet) provided by Cufus, so the user has to insert these himself. For beam properties which are not defined with the standard beam cross-sections, the user can enter these values in the property data file, see section 4.4.1 and appendix C. In future releases the generation of the stress points on the cross-sections of standard beams will be included.

Nodal constraints are defined slightly different to element constraints. These differences consist of the response codes and the user defined area on which the constraint will act. The responses for nodal constraints are limited to displacement constraints. Contrary to element constraints, nodal constraints can act on a single node, pointed to by the cursor or node identification number entered manually, or on the complete model. This can be done both for sensitivity analysis and for structural optimization. The definition of the response code of the nodal constraint is shown figure 31 which, due to limited space, starts at the third menu (column) in figure 28. In a similar way figure 32 (see page 51) shows how the grid[1] def-

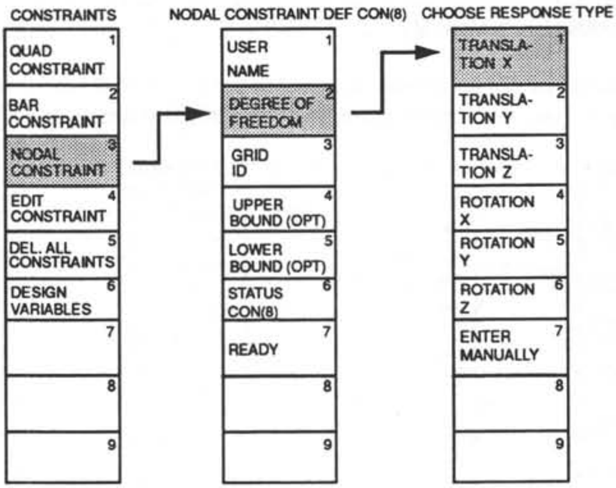


Figure 31: Definition of the response type for a nodal constraint.

inition for the nodal constraint is given. This figures shows that the grid identification numbers

1. Note: In this document the terms *grid* and *node* are mutual interchangeable. This is due to the fact that MSC/NASTRAN uses the term grid for a nodal point, while PATRAN uses the term node for the same entity. A grid in PATRAN is a geometrical point (phase I model, [7]).

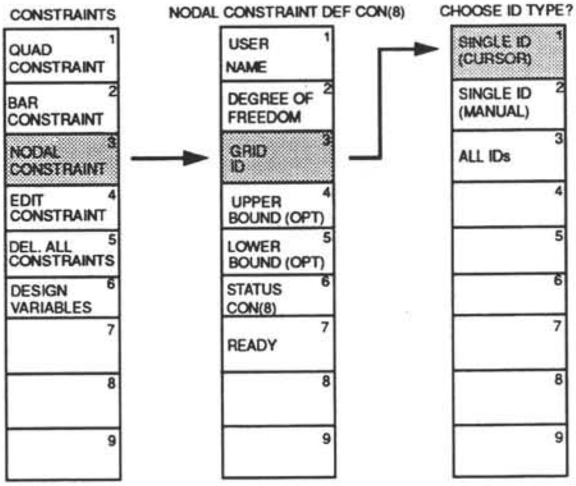


Figure 32: Definition of the response type for a nodal constraint acting on a single node indicated by the mouse cursor.

could be entered in three different ways. For a constraint acting on a single node the node identification number could be entered by pointing to the node in question with the mouse cursor, or by entering the identification number on the keyboard. The third option is used for a constraint definition which acts on the complete finite element model.

The constraints can be edited using the option "EDIT CONSTRAINT" in figure 32. This option allows the user to perform a complete redefinition of the constraints. The constraints are indicated by entering the identification numbers of the constraints, similar to the editing of the design variables. Since it is somewhat difficult to keep track of the constraint identification numbers, it is an useful habit to give the constraints meaningful user names. These names are listed if a constraint identification number is entered for editing. If the name is not correct the constraint can easily be skipped.

Furthermore, there is a very basic way to edit the design variables and the constraints by editing the restart (or dump) file generated by Cufus. After editing the modified dump file can be read again. The format, along with a description, of the dump file is given in appendix C. How to write and read a dump file is described in section 2.5 and 6.1.

The option "DEL ALL CONSTRAINTS" in figure 32 allows the user to remove all constraints. This option can be useful if a complete new set of constraints has to be defined.

5 Additional Post-processing

PATRAN in its standard form does not offer possibilities for pre- and post-processing a MSC/NASTRAN sensitivity analysis. The pre-processing of such an analysis is described in sections 4.3.3 and 4.4. Cufus has the ability to display the results of a sensitivity analysis. The data from MSC/NASTRAN is output in a binary, so called OUTPUT2 file, and an ASCII, so called PUNCH file. This last file is used for post-processing the sensitivity analysis. The Case Control Deck command to generate the punch file is added automatically for the Case Control Deck generation for the sensitivity analysis, see section 4.3.3. This form of post-processing will be discussed in detail in section 5.1.

A special form of post-processing of sensitivity data is the so called "what-if" analysis. By a "what-if" analysis is meant a linear extrapolation of regular output data, such as stresses or displacements, using the sensitivity data. This enables the user to make manual changes to design and have a nearly real time prediction of the structural response. This can be a very useful tool for small adjustments, like for example rounding off the values of the design variables produced by an optimization to values which are commercially available. The user has to keep in mind that all predicted stresses, displacements, etc. are linear extrapolations, so the changes must be sufficiently small to guarantee reasonable accuracy. The "what-if" analysis is described in section 5.2.

The situation for pre-processing and post-processing a structural optimization[1] is similar to that of the pre-processing and post-processing of the sensitivity analysis. The generation of the extension to the Bulk Data Deck and the complete generation of the Case Control Deck is described in section 4.3.3. Besides the regular post-processing of the optimization, Cufus offers also the means to display some of the most important quantities of the optimization by means of two dimensional graphs. In these graphs the history of the maximum constraint violation, the objective function and the design variables can be plotted. The data needed for these graphs can be generated automatically. Section 5.2 describes the special post-processing of a structural optimization in detail.

1. Although shape sensitivity is included, shape optimization is not. MSC/NASTRAN version 67.5 is not capable of performing an shape optimization. MSC/NASTRAN version 68 is able to perform a shape optimization, so it is likely that a future release of Cufus will include the pre- and post-processing means to take full advantage of shape optimization.

5.1 Additional Post-processing: Sensitivity Analysis

Design sensitivity data can be very useful to gain insight in the behaviour of the structure. According to Santos et al. [4, 5] these data provides valuable information about the design variables that are most critical to some constraint, and can supply information concerning the constraints which are most affected by a particular design variable. Therefore it would be a waste of valuable information if this data generated by a structural optimization, were not used. The calculated sensitivity data is also the basis for what-if studies, which can be a powerful tool for the "fine-tuning" of a design.

When using sensitivity analysis and optimization a huge amount of output is generated. The use of these results depends mainly on how the engineer can process this data. Therefore it is considered necessary to use the graphical capabilities offered by PATRAN as much as possible. This way of presenting results can be extremely useful to find the structural elements which have a major influence on the behaviour of the structure.

Besides graphical post-processing of regular data, such as contour plotting stresses and displacements, graphical post-processing of the sensitivity analysis is added also. To enable this kind of post-processing the sensitivity data is translated to a file with a similar format to the stress result files. The sensitivity data is presented in a single way, which means that sensitivity data are also normalized in a single way. In future releases some extra normalizations will be added.

The standard MSC/NASTRAN definition of the sensitivity coefficient is:

$$\frac{\Delta \Psi_i}{\left(\frac{\Delta x_j}{x_j}\right)} = x_i \frac{\partial \Psi_i}{\partial x_j} \qquad (4)$$

Here Ψ_i is the structural response of constraint number i, and x_j design variable number j. The absolute change in the constraint is related to the relative change in the design variable.

The display of sensitivity data can be very helpful to the designer in identifying the most critical structural members, also to show the influence between the different structural members However, it takes some time to really understand exactly what is displayed.

In general it can be stated that in a structure like a cut-out in a pressurized fuselage most changes in structural members result in considerable changes in responses in the other parts of the structure. This implies that "normal" re-design of a structure, which would consist of some kind of fully stressed design cycle (i.e. assuming an independent behaviour of the design variables) will result in a very slowly convergent design process.

The special post-processing can be invoked as shown in figure 33. This menu offers four

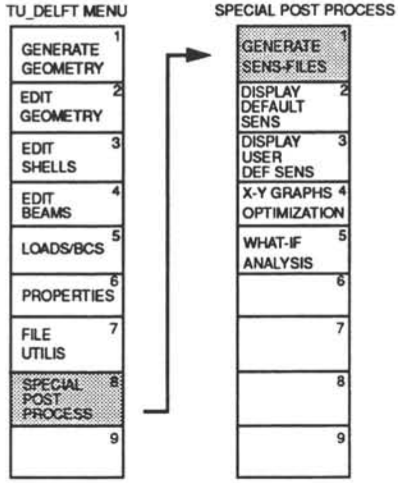


Figure 33: Starting the post-processing of the sensitivity data. The generation of the sensitivity files.

options. The first option is to generate sensitivity files. This means that the results from the PUNCH file [11] are read, and converted in such way that they can be written as element result files to disk. For each constrained component (e.g. Von Mises stresses in the top layer of the shell elements), a separate element result file is written. These files are named sens.Cufus.#, where # stands for a number, starting from 1.

When menu option number 2 "DISPLAY DEFAULT SENS" is chosen now (figure 34), all files with the default name sens.Cufus.# are read. The program detects all available constrained components, and then allows the user to choose the constrained component he wants to view. After that a loop is started in which the user can enter the design variable for which he wants to see sensitivity results. These results are displayed in "fringe" plots[1]. These "fringe" plots are created by extrapolating the element results[2] to the nodes. PATRAN stores the information for the nodes in a separate nodal results file, called patran.nod.#, where # is a number. Because these files have in this particular case no further use, a function is built in to remove them. This function can be found in the "FILE UTILS" menu, see section 6.2.

1. Fringe plots use coloured bands instead of regular contours to indicate the different levels.

2. Element results are calculated in the integration points of the elements.

During the display loop no changes can be made to the viewing window. If a change in viewpoint or magnification is desired, it is necessary to stop the loop, make the changes with the window command, and restart the display loop. The display loop is purposely designed as a "blocking" loop, because in that case the user is able to study the sensitivity results of a relatively large number of design variables in a short time.

The third option in the most right-most menu column in figure 34 is the entry which can be used for displaying non-standard sensitivity data. These data will consist of non-standard normalizations of the sensitivity coefficients. This option is not yet implemented.

5.2 Additional Post-processing: "What-if" analysis

As stated before, the "What-if" analysis is a special type of post-processing sensitivity data. The "What-if" analysis predicts a structural response (stress, strain, force, displacement etc.) based on the sensitivity data. The change in the structural response is approximated by multiplying the derivative of the structural response with respect to the design variable with the change of that design variable, equation (4). The derivative is based on the standard MSC/NASTRAN sensitivity coefficient, equation (4):

$$\Psi'_i \approx \Psi_i + x_i \frac{\partial \Psi_i}{\partial x_j} \frac{\Delta x_j}{x_j} = \Psi_i + \xi_{ij} \frac{\Delta x_j}{x_j} \tag{5}$$

where ξ_{ij} is the standard MSC/NASTRAN sensitivity coefficient of response Ψ_i with respect to design variable x_j (equation 6). The predicted value of the structural response is given by Ψ'_i.

$$\xi_{ij} = \frac{\Delta \Psi_i}{\left(\frac{\Delta x_j}{x_j}\right)} = x_i \frac{\partial \Psi_i}{\partial x_j} \tag{6}$$

It will be clear that the "What-if" analysis is a linear extrapolation of the structural responses from a certain starting point in the design space. This implies that the range where the "What-if" analysis produces reliable accurate results depends strongly on the extent of non-linearity. This non-linearity refers to the statically-indeterminate behaviour of the structure. For example the stress in a structural member will not vary linearly with the inverse of the cross-section of the particular member in a statically indeterminate structure. If the design variables change too much the results will become unreliable. Normally, the allowable change of design variables will be around 20 percent. However, this value is no more than a rule of thumb.

The beginning of the post-processing of the "What-if" analysis is similar to the start of the post-processing of the sensitivity analysis. After invoking the menu option "Special Post Process", figure 34, the option "Generate Sens Files" is chosen. Obviously, if the sensitivity files[1] are already generated, for example for the post-processing of a sensitivity analysis, these files do no longer have to be generated. The preparation of the "What-if" analysis contains the storage of the

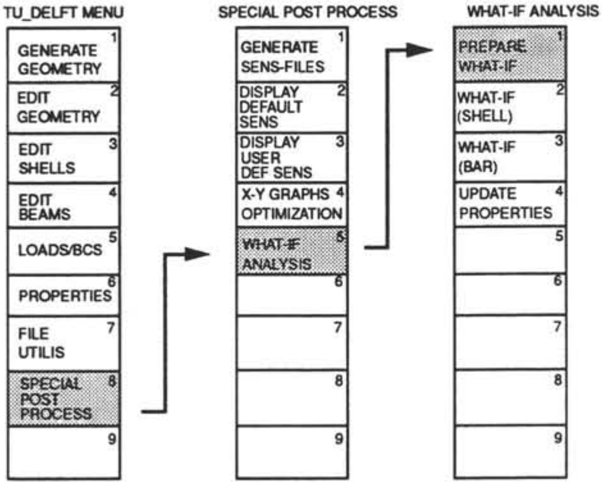


Figure 34: Starting the post-processing of the sensitivity data. The preparation of the "what-if" analysis.

sensitivity data and the stresses in the core memory of the computer. This is a process what may take 5 to 10 minutes, depending on the amount of data to be read. The advantage of in-core storage of these data is the reduction of time required by the analysis itself. A typical change of one of the design variables, resulting in a new stress fringe plot, now takes only 15 seconds, mainly used by the standard PATRAN plotting routine. This implies that the changes in the design results in almost real-time responses. Changes to the model can be made to modify one of the design variables. The design variables to be changed can be chosen by clicking one of the options "What-if (Shell)" or "What-if (Bar)" in figure 34. The program then asks the user to click an element of the design variable to be changed. Cufus retrieves the related design variable number and the current value of the property of the design variable. That current value of the design vari-

1. Sensitivity files are named sens.Cufus.#, where # stands for a number, starting from 1, see section 5.1.

able is shown, and the new value has to be entered by the user. If the new value differs by more than 25 percent of the current value, the user will be notified. If the new value is entered, Cufus will calculate the new stress distribution. When the new responses are calculated and displayed, the program will offer the opportunity to keep the changes in the design. This option is intended to allow the user to make multiple consecutive changes during the "What-if" analysis. This means that the influence of a number of changes can be visualized. If the engineer has reached a satisfactory design in this way, the fourth option in the menu shown in figure 34 enables an automated updating of the finite element model. This option will use the "kept" values of the design variables during the "What-if" analysis as new values for the property definition. The main purpose of this option is to make the changes to the finite element model permanent. The updated finite element model can be used as start for a new finite element calculation to check the accuracy of the "What-if" analysis.

As may be clear from the previous paragraphs in this section, the only responses available for the "What-if" analysis are the stresses in the shell elements. These responses will be the most frequently used responses for a "What-if" analysis. However, it is intended to extend the available responses with stresses of the beam elements and displacements of the nodes.

5.3 Additional Post-processing: Optimization

The fourth option in figure 34, see page 57, is to post-process the data generated by an optimization. The post-processing of regular data like stresses, strains, displacements etc. will not be discussed here, because they are covered in the PATRAN User's Manual [7]. The special post-processing consists of the display in two dimensional graphs of a number of variables against the history of the optimization. The graphs are displayed using a separate part of PATRAN called P/PLOT. The start of this program can be spawned from PATRAN, so the PATRAN session does not have to be interrupted. The variables shown are the objective function, the maximum constraint violation and the design variables, all against the number of design cycles.

These plots can give a valuable insight in the quality of the results generated by the optimization in a very quick way. First of all the display of the maximum constraint violation shows if the optimization has reached a feasible design. Constraints are defined in the following way [12]:

$$g_{2j-1} = \frac{r_j^L - r_j(\underset{\sim}{x})}{|r_j^L|} \leq 0$$
$$g_{2j} = \frac{r_j(\underset{\sim}{x}) - r_j^U}{|r_j^U|} \leq 0 \tag{7}$$

In equation (7) g_{2j} is the $2j^{th}$constraint, r_j^L is the lower bound on the structural response, r_j^U is the upper bound on the constraint and finally $r_j(\underset{\sim}{x})$ is the actual value of the structural response, which is a function of the vector of the design variables $\underset{\sim}{x}$. This equation shows a normalization of the constraint value. The constraints are satisfied when they are smaller than zero and they are considered to be violated when they are larger than zero. For numerical reasons a constraint value below 0.005 is considered to be satisfied. This value can be changed on the DOPTRM-card in the MSC/NASTRAN Bulk Data Deck [11]. For some further comments on the DOPTRM-card, see section 4.3.3.

Another valuable plot is the plot which displays the history of the design variables. This shows clearly which design variables changed the most and if there are variables which are prevented from changing because they have reached their limits. Because the values of the properties of beam elements, which can be the cross-sectional area, one of the second moments of area or torsional stiffness, are of another order of magnitude to the properties of the shell elements, which are thickness, the plots for both types of variables are separated into two graphs. These graphs can lead to a relaxation of some of the limits on the design variables.

The graph of the history of the objective function shows if the objective function has reached a lower value than the initial design. If this is not the case, then it is possible that the initial design was not feasible, resulting in adding some weight to reach a feasible design.

Finally, an option to make similar graphs of the previously mentioned variables at the same time is intended to be included in one of the next releases. This will show if the optimum found, is the correct optimum. It is common practice in optimization that several initial designs are optimized, to ensure that the optimum found is the true one. Displaying the graphs of the different optimizations will enable the user to reach a judgement on the optimum found by MSC/NASTRAN. Furthermore these plots will indicate if the structure has a single or multiple optimum, and show the sensitivity of the optimum with respect to (some of) the design variables. Although this function is not yet implemented in the current release (version 1.0), it is recommended to make these combined graphs by hand. Using the graph data generated by Cufus, consisting of a so-called template file and a xy-data file [7], it is quite easy to modify the template file to view more than one optimization the same time. To perform this modification some knowledge concerning P/PLOT is required.

Unfortunately P/PLOT does not offer the same capabilities to display a user-defined menu as PATRAN does. Therefore manipulating the plots will be done entirely by using the standard menu or by using a small PCL program. This PCL-program can be started by typing

```
!PP()
```

at the command prompt. If this results in an error message try to link the library by typing:

```
!! LIB ADD CF_PP
```

followed by:

```
!PP()
```

This will result in the display of a "menu" in the alpha area. This menu can only be controlled by entering numbers. The options offered will only affect the display of the graphs, no modifications are made to the plotted data. P/PLOT can be ended by typing STOP at the command prompt or by choosing the STOP option in the menu.

6 Miscellaneous Utilities

In this chapter some extra utilities are covered which, due to their nature, cannot be included in one of the previous chapters. These utilities include the reading and writing of restart (or dump) files, and options for the deletion of files generated during a PATRAN/ Cufus session and generally no longer necessary. These utilities are stored under the menu option "FILE UTILS" as showed in figure 35.

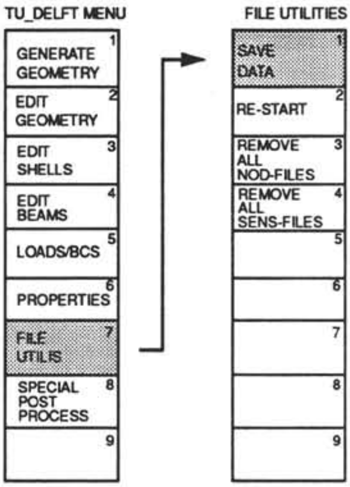


Figure 35: Save the local Cufus data base to disk.

6.1 Writing and Reading Restart Files

Some of the data generated by Cufus can be stored in a restart[1] file. This file is written to disk and can be used in another session. Figure 35 shows how to invoke the menu. The file

1. The term dump file is also used for this file type.

is written in ASCII format. The most important reason for writing the file in that format is that it enables easy debugging. The format of the file is described in appendix C. Along with the information in that appendix it is possible to modify the restart file, in order to edit some variables. However, it is dangerous to modify the file, so be sure that there is always a backup of the original restart file available. Editing a restart file should be used by experienced users only.

Reading a restart file can be done by clicking the second menu option in figure 35. Because reading a Cufus restart file will cause an initialization of the Cufus local data base, a loss of data can occur when this option is clicked by accident. That is the main reason why reading the data base can only be executed when the user has given an explicit confirmation at the keyboard after this option is clicked.

6.2 Deleting Files

There are two options indicated in the left menu column of figure 35, which are meant to delete all files of a certain type. These options are included because during a session many of these files can be generated, while they are not normally necessary during later sessions. For example the third menu option deletes all sensitivity files. These files are named sens. Cufus.#, where # is a number. These files are generated by translating the PUNCH file, so if the PUNCH file is still available these files can be deleted.

Plotting of results uses so-called fringe plots. These plots are normally connected to nodal results, so the element results stored in the files sens. Cufus.# are translated to a nodal results file format. The information in these files is the same as in the sens. Cufus.# files, only the format differs. This implies that the nodal results files are of no use when the PATRAN/ Cufus session is finished. These files are called patran.nod.#, where # is a number. They can be deleted by clicking the fourth menu option in the most left menu column in figure 35.

The file deletion is programmed using PATRAN commands instead of spawning a command for the operating system. This implies that this part of the code is portable over all platforms which are capable of running PATRAN; it is not restricted to Unix machines.

6.3 Creating Assembled Finite Element Models

In typical aircraft fuselage structures the cut-out for a door is often surrounded by other "disturbances" in the structure. For example a passenger door often crosses a window belt. The presence of windows next to the cut-out for the passenger door strongly influences the load path for both the window and the door. This implies that an analysis of a regular fuselage structure with a singular disturbance like a cut-out for a door is no realistic representation of a fuselage structure with multiple disturbances.

Because Cufus in its standard form is only capable of creating finite element models of fuselage structures with a singular cut-out, there is a need to be able to create finite element models with multiple disturbances. A start is made to enable this option. Most of the Cufus capabilities are functional for these assembled models also. Since the development these functions are in a premature stage, most of the functions can only be accessed via the command line. These functions and their arguments will be discussed in section 6.3.2. Furthermore it should be noted that the assembly is more labour intensive than most of the Cufus features.

6.3.1 General Approach

Cufus in its standard form is capable of creating at maximum half of a cut-out. If the models with these cut-outs span only the half of the frame pitch, "building blocks" can be made. Use of standard PATRAN commands allows the user to assemble a model built up from these "building blocks". For example the model shown in figure 36a is built up from the simple models shown in figure 36b.

Creation of the building blocks is completely similar to that of a "normal" Cufus model. After the building blocks are generated, the entities must be renumbered. This can be achieved by entering the command "RENUM" at the PATRAN command line. Both phase 1 (geometric entities) and phase 2 items (nodes and finite elements) must be renumbered. After renumbering the model is saved to a neutral file, and deleted from the current PATRAN data base. This process is repeated for each building block.

When all blocks are finished one of the blocks is read into the data base. If there is already a (part of) the finite element model present in the data base, use the automatic offset of the neutral input id's. (This process is described in the PATRAN User Manual, chapter 29.)

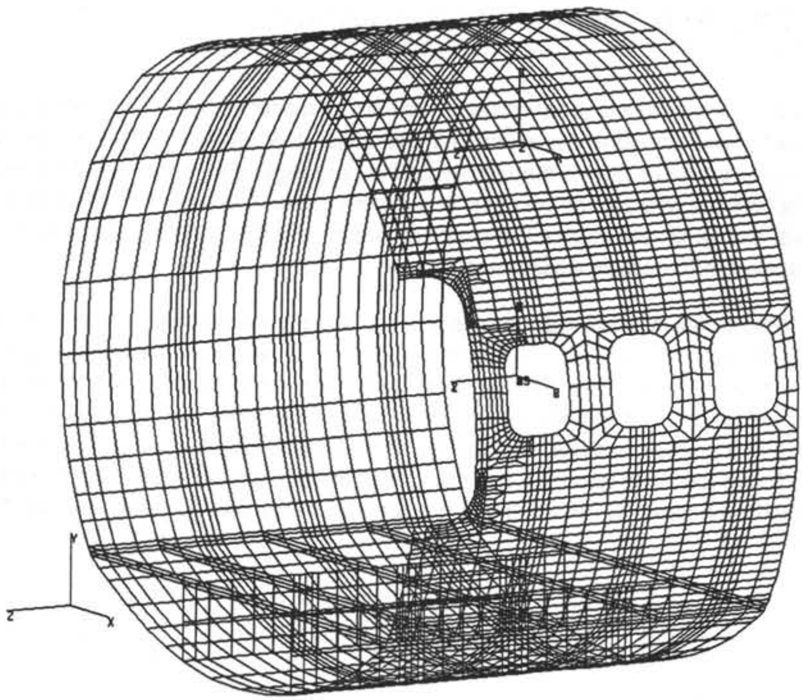


Figure 36a: An assembled model.

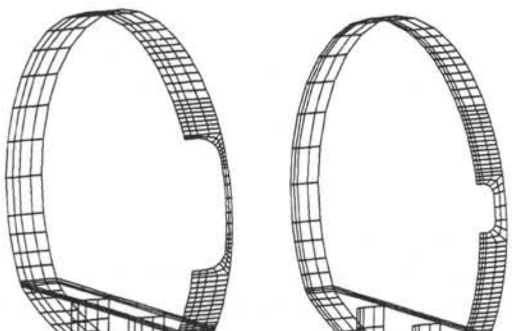


Figure 36b: Building blocks for the assembled model shown in figure 36a.

After reading the building block it can be mirrored and translated using the "name" command. Using this "name" command is described thoroughly in chapter 15 of the PATRAN User Manual [7].

The user has to keep track of the axial position of the building blocks, i.e. the user has to take care that the building blocks are positioned in such way that the outer boundaries of blocks connect to each other. That also implies that on the outer boundaries geometric entities, like lines, patches etc., as well as element entities, like elements, nodes etc., are located at the same position. This can cause serious malfunction of Cufus. Therefore some extra functions are added to remove the duplicate geometry. The duplicate finite element entities can be removed by standard PATRAN functions, see section 4.2.1 and 4.3.1.

All Cufus functions work except the interactive geometry changes to the cut-out. Some of the functions, however, do work but not without some minor flaws. The load application will in most cases apply the MPC's and the corresponding load vector at the wrong frame station. The mesh adaption of the shells will in *some* cases produce bar elements for the stringers in the wrong parametric direction of the patch. In these cases the effect of the mesh adjustment will act in a direction perpendicular to the desired direction. For example clicking the option for mesh refinement in the circumferential direction will result in a mesh refinement in the axial direction. This phenomenon is due to the fact that manipulation with the named components can result in an unforeseen orientation of the patches. Property assignment to (some) of the frames will not work, since not all bar elements at frame locations are recognized as frames any more by the program. For the really experienced user, editing of the dump file can overcome some of these minor problems. The rest of the functions work without any (known) problems.

6.3.2 Assembling Utilities

The assembling utilities are functions meant to enable interactive mesh adaption. Mesh adaption requires information about the geometric entities (lines and patches) on which a finite element mesh is generated. Due to the assembling of the building blocks, the mutual boundaries will have duplicate geometric entities as well as duplicate elements. The duplicate elements can be removed using a standard PATRAN function, the duplicate geometric entities need an user-defined treatment.

There are two functions available, which are specially meant for removing duplicate geometry. The first function is called "**del_same_geom**". This function checks for duplicate lines and/or duplicate patches. Duplicate means that all corner grids of both entities coincide within an user-defined tolerance. The second function is called "**delete_lines_at_z**". The purpose of this function is to remove the lines at a z-coordinate specified by the user. When a model is assembled, lines on the both ends of the fuselage part to be analysed must be removed. This can be achieved by removing the lines by hand or by using this function.

If the function "del_same_geom" is executed, the duplicate elements left can be removed by running the standard PATRAN function as shown in figure 13 (see page 26). Then the actual preparation for the interactive meshing can take place. This preparation is performed by running the function "store_adj_patch". This function stores for every patch its neighbouring patches and lines. Execution of this function can take a large amount of time. For the model shown in figure 36a it took more than 1 hour. If the Cufus data base is saved before finishing the session, this function has to be executed only once for an assembled model.

The functions can be executed by switching PATRAN to the command mode and typing the following at the command prompt:

```
!function(argument_list) ↵
```

The exclamation sign is necessary to indicate that the command is a PCL-function. The argument list is specific for each function. The arguments are listed for each function in table 4.

Function name	**argument s**	**description**
del_same_geom	(tol, job)	Function to delete geometric entities which have coinciding corner grids. tol = Tolerance in mm for line checking, advised value: tol =1.0 (real) job = Control what to do (integer): job = 1 or job = -1 Only line checking job = 2 or job = -2 Only patch checking job = 3 or job = -3 Both line and patch checking In case job is negative the meshes on the deleted entities are also deleted. Advised value: job = -3.
delete_lines_at_ z	(tol)	Function to delete all lines at a certain, user-defined z-coordinate. Z-coordinate is entered by clicking a grid point. tol = Tolerance in mm for line checking, advised value: tol =1.0 (real)

Table 4: Argument description functions for model assembly.

Function name	argument s	description
store_adj_patch	(nc, gen)	Function to store all surrounding lines and patches for each patch. nc = Name of named component to be checked. Members of other named components will be skipped. If nc = "NONE" all lines and patches will be checked. (Character) Advised value: "NONE" gen = Indicates if the function is executed for a standard or an assembled model. For an assembled model the value should be: gen = TRUE. (Logical) Advised value: gen = TRUE.

Table 4: Argument description functions for model assembly. (Continued)

7 An Illustrative Example

An example is given in appendix E. This example is a design session which starts from a few dimensions of the fuselage and the cut-out and ends with an optimized design. The figures in the examples are sometimes slightly modified, e.g. contour plots instead of fringe plots, for reasons of clarity or hardcopy device limitations. However, the functionality of the example is exactly the same as it is in reality.

In this example not every key stroke is included, but it is supposed to be clear enough to serve as an example session. The use of the example shown in this appendix is especially recommended as a first acquaintance with Cufus.

References

1 M.C.Y. Niu: *Airframe Structural Design.* **Conmilit Press Ltd., Hong Kong, 1990.**

2 E. F. Bruhn: *Analysis of and Design of Flight Vehicle Structures.* **Tri-State Offset Company, Cincinnatti, Ohio, U.S.A., 1965.**

3 P. Kuhn: *Stresses in Aircraft and Shell Structures.* **McGraw-Hill Book Company, Inc. New York, 1956.**

4 J.L.T. Santos, M.M. Godse and Kuang-Hua Chang: *An Interactive Post-Processor for Structural Design Sensitivity and Optimization: Sensitivity Display and What-If Study.* **In:"** *Computers and Structures"*, **vol. 35, no. 1, pp. 1-13, 1990.**

5 J.L.T. Santos and K.K. Choi: *Integrated Computational Considerations for Large Scale Structural Design Sensitivity Analysis and Optimization.* **Discretezation Methods and Structural Optimization Procedures and Applications, Proceedings of a GAMM-seminar, Oct. 5-7, 1988, Siegen, FRG.**

6 *PATRAN Command Language (PCL) Guide.* **Release 2.5, PDA Engineering, Costa Mesa, California, USA, 1990.**

7 *PATRAN Plus User Manual.* **Release 2.5, PDA Engineering, Costa Mesa, California, USA, 1990.**

8 M.E. Heerschap: *Programmers Manual CuFus, Quick cut-out FUselage design.* **To be published.**

9 M.E. Heerschap: *Simplified Finite Element Representation of Fuselage Frames with Flexible Castellations.* **Computers and Structures, vol.47, No. 1, pp. 19-32, 1993.**

10 *PAT/MSC/NASTRAN Application Interface.* **Release 3.0, PDA Engineering, Costa Mesa, California, USA, 1991.**

11 *MSC/NASTRAN User's Manual version 67.* **The MacNeal-Schwendler Corporation, Los Angeles, USA, 1991.**

12 G.J. Moore: *User's Guide MSC/NASTRAN Design Sensitivity and Optimization Version 67.* **The MacNeal-Schwendler Corporation, Los Angeles, USA, 1993.**

13 *Technical Handbook 3 Strength Data.* **Fokker Aircraft BV, dept. EBDI/H, 1992.**

Appendix A: Operational Requirements

A.1 General system set-up

For the complete design system three different computers are used at the same time. The file- and compute-server runs PATRAN and Cufus and the two translators NASPAT and PATNAS. The results are displayed at a local machine. This local machine and the file-server are placed in a local SUN network and share the disks. Therefore the disk usage on these two machines is very transparent, i.e. no file transfer is required. In the SUN-network a multi-processor computer (Sparc 1000) is also included. This multi-processor computer runs MSC/NASTRAN. This computer can only be used through batch-processing. Figure A.1 shows the system set-up. As well as the installation of various programs, this appendix will deal with, the system set-up in somewhat more detail.

A.2 How to run PATRAN

In order to run PATRAN on one of SUN Unix stations, a number of settings should be added or changed. First of all PATRAN needs a SUN-station which runs Open Windows, preferably a station with a colour monitor. PATRAN itself is run on one of the file servers (dutlfs4) and uses the local SUN station as a so-called X-server for displaying the results. Before this can be done the following lines should be added to the .login-file:

```
if ('hostname' == "dutlfs4") then
      set path=($path /patran/bin)
      alias remotePatran    'source /vol/diversen/bin/remotePatran'
else
      setenv     DISPLAY "'hostname':0.0"
endif
```

To make these changes effective it is necessary to log out and log in again, or type at the command line:

Appendix A: Operational Requirements

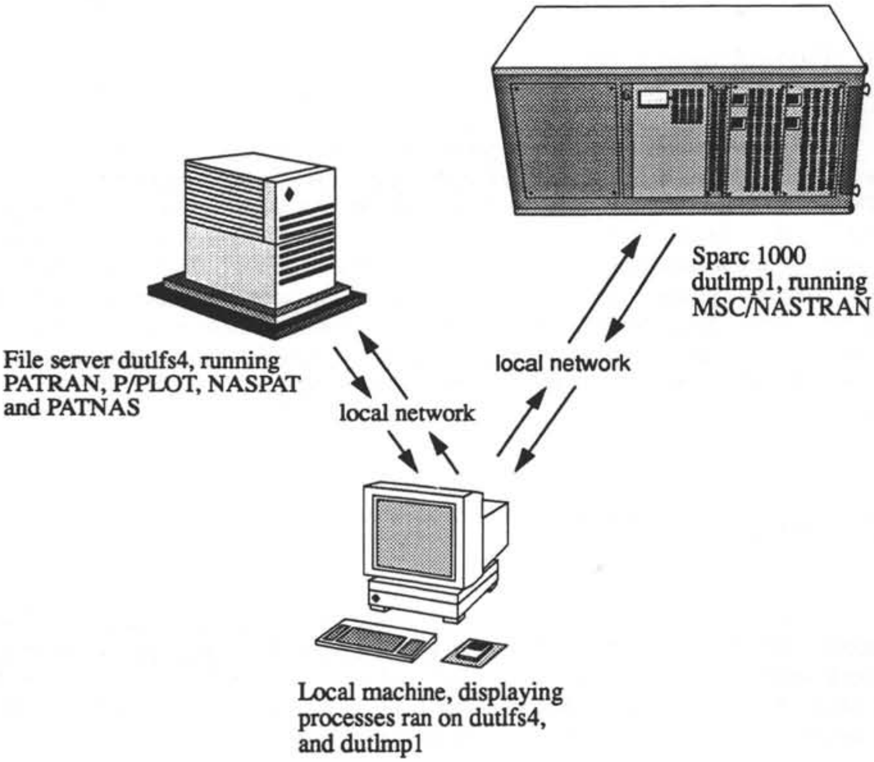


Figure A1:The three necessary computer systems

```
source .login ↵
```

It is assumed here that the settings for accessing Open Windows are already correct. If Open Windows does not starting correctly, contact the system operator. To start PATRAN, log in to the computer. Start Open Windows by typing the following line at the command prompt:

```
openwin ↵
```

Appendix A: Operational Requirements

If there are no shells opened by default, open a shell-tool or a command tool. In one of these tools the next command has to be typed:

```
xhost dutlfs4 ↵
```

Then a remote login to the file server dutlfs4 is required by typing:

```
rlogin dutlfs4 ↵
```

The computer will now ask for a password. After entering the password, the user can change directory, if necessary, and access PATRAN by typing:

```
remotePatran ↵
```

This script will ask for the name of the local host, i.e. the machine which is being used to display the results. After that PATRAN can finally be started by entering:

```
patran ↵
```

If patran starts, it will ask for a device mnemonic. These device mnemonics can be "bat" for the batch-mode and "xwn" for the interactive graphical display of PATRAN. For example:

```
ENTER DEVICE MNEMONIC, "?", OR "STOP": xwn ↵
```

The commands which are given here should be executed each session, with the exception of the changes in the .login file. During a session PATRAN can be executed again by entering "patran" at the command prompt of the shell which is logged in to dutlfs4.

A.3 Installing Cufus

In order to start the design system Cufus within PATRAN the following files should be available:

`patuser.emn`	menu system
`Cufus.pcl or Cufus.plb`	Program of Design System Cufus
`startup.pcl`	Initial settings PATRAN
`demo.dat`	Demo file for key dimensions
`prop.dat`	Demo file with property data

Normally the start-up file executes a number of commands which will result in a ready to use design system, provided that all files are available in the current directory. The menu system can be changed by the user. How to change this menu is described in the PATRAN PCL manual [6].

The file startup.pcl can also be changed by the user, however it should be noted that not all available commands in PATRAN which can be placed in this file are also available for P/PLOT. This can lead to some "error beeps" while starting P/PLOT. These "error beeps" are in general quite harmless. Another problem is that the start-up file is only read when PATRAN is in the command mode. This implies that the user, at the beginning of the session, after a data base has been opened, has to type "C" (command mode). Then the file startup.pcl is automatically read, and the command mode can be left by typing "E" (end). If an error message is displayed saying that a certain function can not be accessed, the design system has probably not been loaded into PATRAN. In this case start the command mode (by typing "C"), and type at the prompt:

!!LIB ADD CUFUS.PLB ↵

or

!!INPUT CUFUS.PCL ↵

This depends on which file (extension plb or pcl) is available to the user. The file cufus.plb is a compiled version of cufus.pcl. For modifying the source code the uncompiled version, cufus.pcl is necessary.

A.4 Accessing MSC/NASTRAN

The design system Cufus uses MSC/NASTRAN. MSC/NASTRAN is run at the multi-processor computer Sparc 1000. This computer can only be accessed through a batch system. MSC/NASTRAN jobs can be submitted by entering the following command at the command prompt of the local computer:

```
qsub -q dutlmpl job-name ↵
```

MSC/NASTRAN should be started by sending a job file (with the name job-name in the example) to the batch system on the SParc 1000. Without discussing this in detail, a example of a generic job file is given below:

```
# Sun/SPARC job for NQS v3.21 & CERN v2.5
#
# NQS default parameters (see man qsub)
# ------------------------------------
# @$-eo
# @$-nr
# @$-o put_here_your_job_name.log
# @$-q batch
# @$-r NastranV68
# @$-s /bin/csh
# @$
# ------------------------------------
#
nastran put_here_your_job_name
#
exit 0
```

- Further information can be found in the lecture notes for lr-25 inl. FEM.

Appendix B: The Menu System

B.1 Overview Menu System

In this appendix the menu system will be described. Figure B1 shows the complete menu system used in Cufus. It includes the part of the menu system contained in the file patuser.emn as well as included in the source code of the program. If PATRAN is started and the menu system is not available, then check if the file patuser.emn is available in the current directory. If this file is not available, copy or move the file to the current directory and restart PATRAN.

The part of the menu system embedded in the source code disappears from the screen when user input from the alpha-area is required. Normally the menu switches back to another menu from the file patuser.emn. While this is confusing, the right menu will appear again when the user input is finished by hitting the "Return-key".

B.2 Modifying the Menu

This section will briefly describe how to modify the menu included in the file patuser.emn. A more extended extension can be found in the PATRAN PCL manual [6]. The generic structure of the menu file is:

```
MENU menu-name
    ITEM item-name
        SMENU menu-name or PCL command
    ITEM item-name
        SMENU menu-name or PCL command
        •
        •
        •
        •
    ITEM item-name
        SMENU menu-name or PCL command


MENU menu-name
    ITEM item-name
```

Appendix B: The Menu System

```
        SMENU menu-name or PCL command
ITEM item-name
        SMENU menu-name or PCL command
```

The words written here in bold capitals are keywords, i.e. these words must be used unchanged. The words written in underlined italics are user defined.

A menu is built up of maximum eight items and is preceded by the keyword "MENU". The name coming after this keyword is the title of the menu. The items of the menu are indicated by the keyword "ITEM". The name following that keyword is the name of the item and displayed in the menu box. An item can contain PCL commands, standard PATRAN commands or a pointer to a sub menu. PATRAN commands can be typed in the same way as they would be entered in the alpha area. PCL commands should be preceded by the keyword "PCL" and followed by an exclamation sign (!) which is the same as in a PCL program. Sub menus can be called by the keyword "SMENU" followed by the sub-menu name. The sub-menu must be also included in the file patuser.emn.

As example the first part of the Cufus menu file is given here:

```
MENU TU_DELFT MENU
      ITEM GENERATE GEOMETRY
           SMENU GENERATE GEOMETRY
      ITEM EDIT GEOMETRY
           SMENU EDIT GEOMETRY
      ITEM EDIT SHELLS
           SMENU EDIT SHELLS
      ITEM EDIT BEAMS
           SMENU EDIT BEAMS
     ITEM LOADS/BCS
            SMENU LOADS/BCS
      ITEM PROPERTIES
           SMENU PROPERTIES
      ITEM FILE UTILS
           SMENU FILE UTILITIES
      ITEM SPECIAL POST PROCESS
           SMENU SPECIAL POST PROCESS

$
$ Sub menu for file utilities
$
 MENU FILE UTILITIES
     ITEM SAVE DATA
           PCL !DUMP_DATA()
     ITEM RE-START
           PCL !RE_START()
     ITEM REMOVE ALL NOD-FILES
           PCL !REMOVE_FILE("patran.nod.#")
     ITEM REMOVE ALL SENS-FILES
           PCL !REMOVE_FILE("patran.nod.#")
```

Appendix B: The Menu System

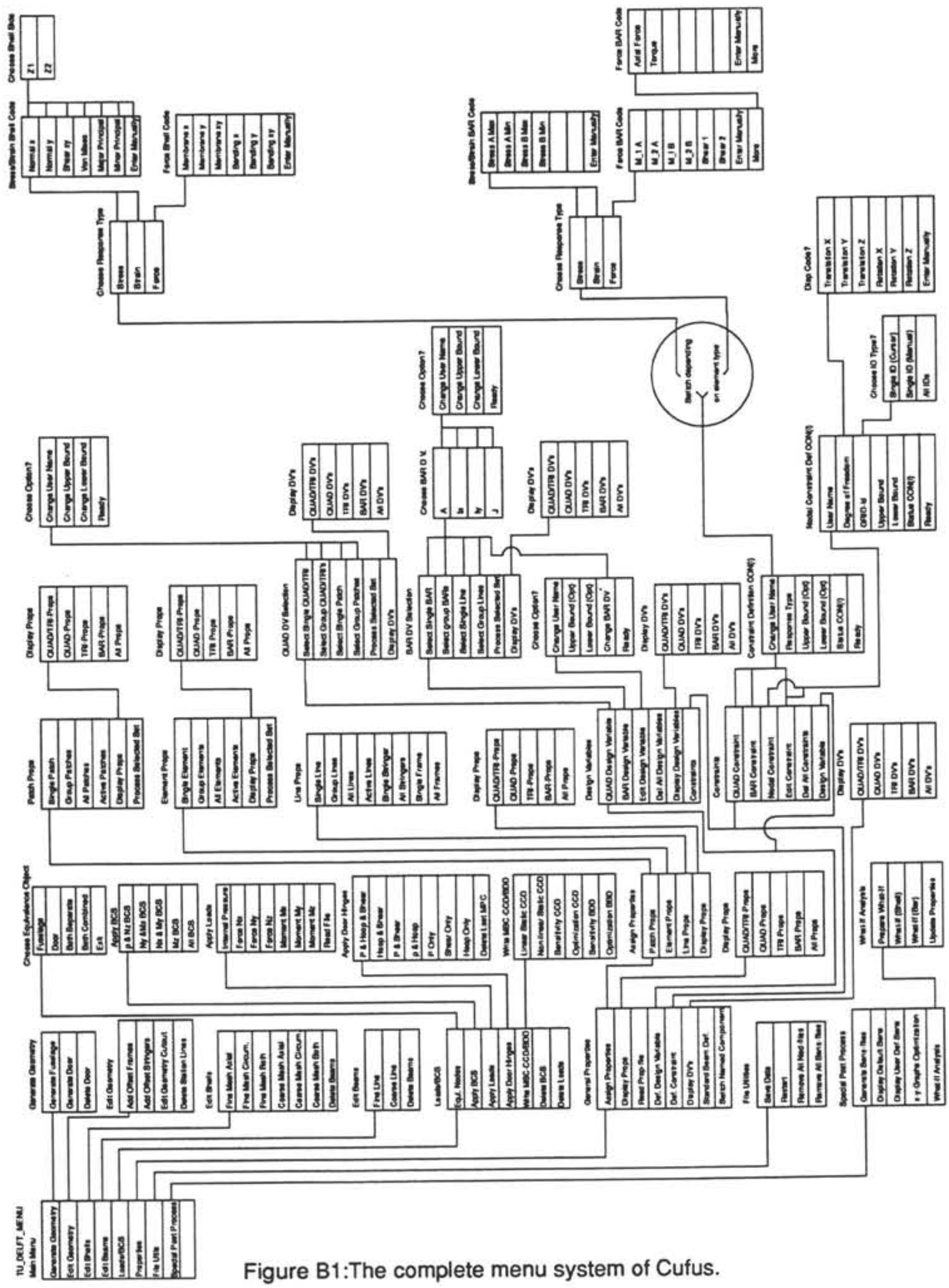


Figure B1:The complete menu system of Cufus.

Appendix C: The Required Files for Cufus

C.1 Geometric input file

A short description of the file containing the geometric data is given below. A more extensive description of some parameters is given in the paragraphs 3.1, 3.2 and 3.3.

Use of the input file is optional, as data can also be entered by the keyboard. In general all files may include comment lines. The first character of a comment line must be a dollar sign ($). Blank lines are *not* interpreted as comment lines, but as data lines.

Some values of parameters are indicated as "default". This implies that for some reason a parameter cannot be read from the data file this value will be used. It does ***not*** mean that this value can be omitted.

This file has a fixed format. All data values can use the first 10 positions. After each tenth column comment can be inserted. If comment is given after the tenth position, it is not mandatory to start the comment by a dollar sign.

Description of the contents of the file:

Radius of fuselage [mm]	**real**	**10 positions**	**(F10/E10)**
Number of frames	**integer**	**10 positions**	**(I10)**
Position of frames [mm]	**real**	**10 positions**	**(F10/E10)**
Repeat this card for each frame			
Number of stringers	**integer**	**10 positions**	**(I10)**
Position of stringers [mm]	**real**	**10 positions**	**(F10/E10)**
Repeat this card for each stringer			
Cut-out type	**string**	**80 positions**	

Valid entries are: Rectangular cut-out with rounded corners
Elliptical cut-out
User defined cut-out

The keywords are case insensitive. The keyword will only be scanned for the first character, which implies that one of the characters R, E or U will suffice. It is mandatory that the keywords start at the first column. The option User defined cut-out is not yet implemented.

Height of cut-out [mm] **real** **10 positions (F10/E10)**

Use for rectangular cut-out with rounded corners and elliptical cut-outs. It is the total height of the door.

Position of cut-out [mm] **real** **10 positions (F10/E10)**

The position of the lower boundary of the cutout. Measured along the contour of the fuselage in circumferential direction.

Width of cut-out [mm] **real** **10 positions (F10/E10)**

Use for rectangular cut-out with rounded corners and elliptical cut-outs. This is the width of the half cut-out.

Radius of the corner of cut-out [mm] **real** **10 positions (F10/E10)**

Use only for rectangular cut-out with rounded corners. Skip this card for other cut-out types.

Starting angle of fuselage [degrees] **real** **10 positions (F10/E10)**

Use only 0.0 for a 360 degrees model.

Ending angle of fuselage [degrees] **real** **10 positions (F10/E10)**

Use only 360.0 for a 360 degrees model.

Starting offset of fuselage [mm] **real** **10 positions (F10/E10)**

Extra length of fuselage before first frame.

Ending offset of fuselage [mm] **real** **10 positions (F10/E10)**

Extra length of fuselage after last frame.

Presence of floor **string** **80 positions (A80)**

Valid entries are: Floor present
No floor present. (default)

The keywords are case insensitive. The keyword will only be scanned for the first character, which implies that one of the characters F or N will suffice. It is mandatory that the keywords start at the first column. If the keyword is not recognized, no floor will be generated.

Position of floor [mm] **real** **10 positions (F10/E10)**

Only valid if the floor is present. The position of the floor is measured from the origin of the coordinate frame of the primary fuselage bubble, see figure B1. Upwards is positive.

Appendix C: The Required Files for Cufus

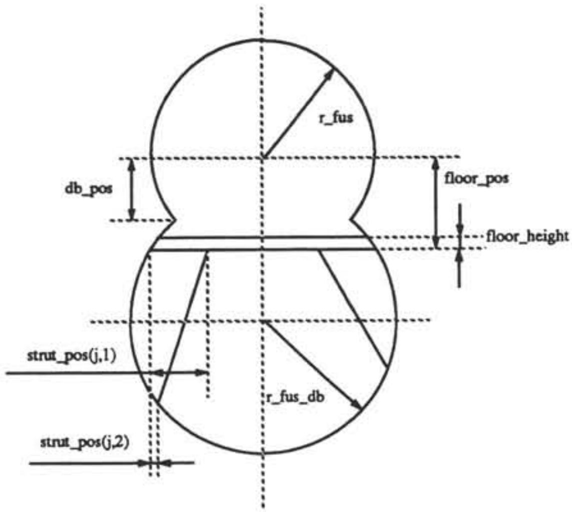


Figure C1:Some of the required dimensions for a double bubble fuselage with a floor

height of floor [mm] **real** **10 positions** **(F10/E10)**

Only valid if the floor is present.

Number of struts **integer** **10 positions** **(I10)**

Only valid if the floor is present. Use 0 if no floor struts are present. Note that for the generation of crash beams floor struts must be present.

Location of struts at floor position [mm] **real** **10 positions** **(F10/E10)**

Location of struts at fuselage position [mm] **real** **10 positions** **(F10/E10)**

Repeat last two cards for each strut. The **locations** of the struts are explained in figure B1.

Number of crash beams **integer** **10 positions** **(I10)**

Only valid if the floor is present. Use 0 if no crash beams are present.

Number of strut to be turned into a crash beam **integer** **10 positions** **(I10)**

Repeat this card for each crash beam.

Presence of double bubble **string 80 positions (A80)**

Only valid for 360 degrees models.

Valid entries are: Double bubble present.

No double bubble present. (default)

The keywords are case insensitive. The keyword will only be scanned for the first character, which implies that one of the characters D or N will suffice. It is mandatory that the keywords start at the first column. If the keyword is not recognized, no double bubble fuselage will be generated.

Radius of double bubble fuselage [mm] **real 10 positions (F10/E10)**

Only valid if the double bubble fuselage is present.

Position of double bubble fuselage [mm] **real 10 positions (F10/E10)**

Only valid if the double bubble fuselage is present. See figure B1 for the definition of this parameter.

Type of double bubble fuselage **string 80 positions (A80)**

Only valid if the double bubble fuselage is present.

Valid entries are: Extended bubble.

Retracted bubble. (default)

The keywords are case insensitive. The keyword will only be scanned for the first character, which implies that one of the characters E or R will suffice. It is mandatory that the keywords start at the first column. If the keyword is not recognized, a retracted bubble will be generated. See for the definition figure 10, see page 18.

C.2 Load input file

Use of the next file, containing (some of) the load case data, is optional. The format is fixed. Each line contain three values, first two integers and the third must be real. For each value 10 positions may be used. After the 30th column comment may be inserted. Just as in the previous file, comment can be inserted anywhere, *if* the character in the first column is a "$" (dollar sign). The data may *not* be separated by commas or other separators.

Description of the contents of the file:

Load type, Load case set-id, Magnitude of load (I10, I10, F10)

This card can be repeated for each load case. Description of load type:

1.......internal pressure

2.......Nx, shear load in global X-direction

3.......Ny, shear load in global Y-direction

4.......Nz, normal load in global Z-direction

5.......Mx, bending moment along global X-direction

6.......My, bending moment along global Y-direction

7.......Mz, twisting moment along global Z-direction

C.3 Property data library file

Use of the next file, containing the property data, is optional. The format is only partly fixed. First of all the materials are defined. This definition is the same as the material definition in PATRAN (free format, use of commas as separator mandatory) for example:

PMAT, 1, ISO, 72000., 27067.67, 0.33, 2.7E-6

The element properties can then be defined. These properties are according to the definition in the MSC/NASTRAN Users Manual Vol. I description of PSHELL and PBAR cards [11]. The format used: The first 10 characters are used for the definition of the property id, after that the following cards can be given (free format, use of comma's as separator mandatory):

QUAD, MID1/ T/ MID2/ (12I/T3)/ MID3/ (TS/T)/ NSM/ (Z1,Z2)/ MID4

or

BAR, MID1/ A(A)/ I1(A)/ I2(A)/ I12(A)/ J(A)/ NSM(A)/ (Ci(A), Di(A),

Ei(A), Fi(A))/ SO/ (X/XB)/ (A, I1, I2, I12, J, NSM)/ (Ci, Di, Ei,

Fi)/ (K1, K2)/ (S1, S2) etc.........

Variables mentioned in these lines are fully described in he MSC/NASTRAN Users Manual Vol. I, PSHELL and PBAR cards[11].

As in the previous file, comment can be inserted anywhere, *if* the character in the first column is a "$" (dollar sign).

C.4 Re-start or Dump File

The re-start file (or dump file) is a file which the user normally does not have to edit. However, in some cases, especially during programming, it can be useful to view or edit the contents of this file. It is written in ASCII format to enable easy editing and debugging. The file is written entirely in fixed format. Comment can *not* be inserted.

Description of the file:

Radius of fuselage [mm] — **real** — **10 positions (F10/E10)**

Number of frames — **Integer** — **10 positions (I10)**

Position of frames [mm] — **real** — **10 positions (F10/E10)**
Repeat this card for each frame.

Number of stringers — **Integer** — **10 positions (I10)**

Position of stringers [mm] — **real** — **10 positions (F10/E10)**
Repeat this card for each stringer

Cut-out type — **Integer** — **10 positions (I10)**
Valid entries are: 1 = Rectangular cut-out with rounded corners
2 = Elliptical cut-out

Height of cut-out [mm] — **real** — **10 positions (F10.3)**

Position of cut-out [mm] — **real** — **10 positions (F10.3)**

Width of cut-out [mm] — **real** — **10 positions (F10.3)**

Radius of the corner of cut-out [mm] — **real** — **10 positions (F10.3)**
Only in case cut-out type equals 1.

Starting angle of fuselage [degrees] — **real** — **10 positions (F10.3)**

Ending angle of fuselage [degrees] — **real** — **10 positions (F10.3)**

Starting offset of fuselage [mm] — **real** — **10 positions (F10.3)**

Appendix C: The Required Files for Cufus

Ending offset of fuselage [mm]	**real**	**10 positions**	**(F10.3)**
Number of non-zero elements in array p_info	**integer**	**9 positions**	**(I9)**

Definition in [8].

Value of element array p_info	**integer**	**4 positions**	**(I4)**

Repeat this card for each non-zero element of this array. Definition in [8].

Number of non-zero elements in array p_adj	**integer**	**9 positions**	**(I9)**

Definition in [8].

Value of element array p_adj	**integer**	**4 positions**	**(I4)**

Repeat this card for each non-zero element of this array. Definition in [8].

Load type	**integer**	**5 positions**	**(I5)**
Load set identification number	**integer**	**5 positions**	**(I5)**
Load magnitude	**real**	**15 positions**	**(E15.7)**

Repeat these cards for a fixed number of load cases. The number of load cases is set to 100. See for a more detailed description [8].

Current load case	**integer**	**5 positions**	**(I5)**
Number of used design variables	**integer**	**9 positions**	**(I9)**
Design variable number	**integer**	**5 positions**	**(I5)**
Property identification number	**integer**	**5 positions**	**(I5)**
Property field number	**integer**	**5 positions**	**(I5)**
Upper bound	**real**	**10 positions**	**(F10.3)**
Lower bound	**real**	**10 positions**	**(F10.3)**
Name of the design variable	**character**	**8 positions**	**(A8)**

Repeat these six cards for each design variable.

Number of used constraints	**integer**	**9 positions**	**(I9)**
Constraint identification number	**integer**	**5 positions**	**(I5)**
Grid or element identification number	**integer**	**5 positions**	**(I5)**
Constrained component	**integer**	**5 positions**	**(I5)**
Used property identification number	**integer**	**5 positions**	**(I5)**
Upper bound	**real**	**10 positions**	**(F10.3)**
Lower bound	**real**	**10 positions**	**(F10.3)**
Name of the constraint	**character**	**8 positions**	**(A8)**
Name of the response type	**character**	**8 positions**	**(A8)**

Repeat these eight cards for each constraint.

Appendix C: The Required Files for Cufus

Number of used entries of sensitivity pointers	**integer**	**9 positions**	**(I9)**
Sensitivity pointer	**integer**	**10 positions**	**(I10)**

Repeat these eight cards for each sensitivity pointer.

Number of used entries of geometry modification	**integer**	**9 positions**	**(I9)**

Used for interactive geometry modification, see section 3.4.

Patch number for geometry modification	**integer**	**10 positions**	**(I10)**

Repeat this card for each non-zero item in this array.

Number of used beam eccentricities	**integer**	**9 positions**	**(I9)**

Beam identification number	**integer**	**10 positions**	**(I10)**
Beam eccentricity x	**real**	**10 positions**	**(F10.3)**
Beam eccentricity y	**real**	**10 positions**	**(F10.3)**

Repeat this card for each non-zero item in this array. Not yet functional.

Presence of floor	**string**	**80 positions**	**(A)**

Valid entries are: Floor present.
No floor present.

Position of floor	**real**	**10 positions**	**(F10.3)**

only valid if the floor is present.

Number of struts	**integer**	**10 positions**	**(I10)**

only valid if the floor is present.

Location of struts at floor position [mm]	**real**	**10 positions**	**(F10.3)**
Location of struts at fuselage position [mm]	**real**	**10 positions**	**(F10.3)**

Repeat last two cards for each strut. The **locations** of the struts are explained in figure B1.

Number of crash beams	**integer**	**10 positions**	**(I10)**

Only valid if the floor is present.

Number of strut to be turned into a crash beam	**integer**	**10 positions**	**(I10)**

Repeat this card for each crash beam.

Presence of double bubble	**string**	**80 positions**	**(A80)**

Only valid for 360 degrees models.
Valid entries are: Double bubble present.
No double bubble present.

Radius of double bubble fuselage [mm] **real** **10 positions** **(F10/E10)**

Only valid if the double bubble fuselage is present.

Position of double bubble fuselage [mm] **real** **10 positions** **(F10/E10)**

Only valid if the double bubble fuselage is present. See figure B1 for the definition of this parameter.

Type of double bubble fuselage **string** **80 positions** **(A80)**

Only valid if the double bubble fuselage is present.

Valid entries are: Extended bubble.
Retracted bubble. (default)

The terms used in the description in the file are fully explained in the comment lines of the program itself and in [8]. The functions used for the writing and reading of the restart file are "DUMP_DATA" and "RE_START" [8].

Appendix D: Status of Cufus Version 1.0

D.1 The Capabilities of Cufus in the Current Version

General capabilities

- Quick modelling, i.e. less than 1 hour.
- User friendly menu driven.
- Input by file, by keyboard or by menu, in almost all cases from choice.
- Possibility to interrupt the design session by saving all necessary data.
- Data necessary to increase performance of the program is stored in-core.
- Visualization the results of the sensitivity calculation.
- Post-processing of one or multiple structural optimizations by visualizing the optimization history.
- An implementation of an user-drive "what-if" analysis.
- It must be possible to generate fuselage sections up 360 degrees with a complete cut-out. This implementation follows a few intermediate stages, i.e. a quarter of a cut-out, half of the cut-out and finally the complete cut-out.

Modelling capabilities

- Variables of geometry:
  - radius of fuselage
  - radius of double bubble
  - number of frames
  - position of frames
  - number of stringers
  - position of stringers
  - height of the cut-out
  - width of the cut-out
  - radius of the corner of the cut-out
  - type of the cut-out, rectangular with rounded corners, elliptical or circular
  - starting angle of the fuselage
  - ending angle of the fuselage
  - starting axial offset of the fuselage
  - ending axial offset of the fuselage
  - presence of floor
  - height of floor
  - position of floor struts
  - presence of crash beams
- Initial model is built on basis of the parameters described above. These values can be entered by file or by keyboard.
- A quarter or a half of the door can be modelled.
- It is possible to model up to 360 degrees of the fuselage structure.

- Mesh adapting takes place completely interactively.
- During mesh adaption the connectivity of the elements is ensured through automatic modification of surrounding elements, including bar-elements for stringers and frames.
- Option to extend or retract the double bubble part of the fuselage.
- A library of materials, thicknesses and beam properties can be defined and read by the program.
- An library of standard beam cross-sections is available. This data is stored in parametric format, so the user can define non-standard dimensions for one of these cross-sections.
- Interactive, mouse controlled property assignment.
- The geometry of the cut-out can be modified (within certain limits) after the model has been generated. The connectivity of the elements is guaranteed, and possible mesh adaptions are also saved.
- Properties can be assigned to entities listed in table 3, page 41.
- The definition of design variables is very similar to the property assignment.
- Design variables are applicable to sensitivity analysis as well as structural optimization.
- Graphical display of design variables.
- Interactive constraint definition for as well sensitivity analysis as structural optimizations.
- Automatic generation of necessary constraints, i.e. the user has only to define a single constraint and then indicate on which are it is applicable, not every constraint has to be defined explicitly.

Capabilities boundary conditions and loads

- Application of symmetric and anti-symmetric boundary conditions for fuselage sections up to 360 degrees.
- For each load case the program will search for the right set identification number of the boundary conditions.
- Multiple load cases can be defined for a single run. The number of load cases is (arbitrarily) limited to 100.
- Loads can be applied through file or interactive input. The following loads can be applied:
- internal pressure
- forces in x, y & z direction acting on a complete cross-section of the fuselage.
- moments along x, y & z direction, acting on a complete cross-section.

Special MSC-NASTRAN capabilities

- For a linear static, non-linear static, a sensitivity analysis and a structural optimization, a Case Control Deck can be generated automatically.
- For as well sensitivity analysis as structural optimization the required extension to the Bulk Data Deck is generated.
- Data recovery and processing from sensitivity analysis. This enables post-processing of the results of the sensitivity analysis.
- Performing a "What-if" analysis.
- Post-processing of an optimization.

D.2 Future Enhancements

This section gives a global view of the items which are under development. This list of future enhancements is not a very rigid list, it will be modified due to desirability of some options, or due to problems with implementation.

New or extra general items

- Migration from PATRAN Plus version 2.5 to PATRAN P3.
- Data generated by previous finite element runs should be converted easily, so it is to use for a new PATRAN-session.
- Implementation of an on-line help-system.
- Further development of the user interface.

D.3 Bug List

This bug list describes the errors which are known to exist.

Modelling

- Generation of the door itself only works for rectangular cut-outs with rounded corners, and for models with a quarter of cut-out.
- Cut-outs can not be generated correctly if the cutout spans less than 2 stringer pitches in circumferential direction.
- On assembled models several malfunctions of functions (can) occur. For a more detailed description see section 6.3.
- Not all load case options are tested.
- Generation of boundary conditions not prepared for fuselage sections more than 90 degrees.

Appendix E: Illustrative example

E.1 Introduction

In this appendix a more or less typical example of a circular pressurized fuselage with a cut-out for a passenger door will be treated. The appendix is divided into several sections, in agreement with the different steps in the design process.

The structure to be designed in this appendix is a circular fuselage with a radius of 1400 mm, subjected to a differential pressure of 0,05 MPa. The size of the cut-out is 1300 mm height (total), 800 mm width (total) and the radius of the corner of the cut-out is set to 100 mm. The frame pitch is 500 mm, the stringer pitch is set to 150 mm over the section to be analysed. The section to be analysed spans 90 degrees. The design constraints are: maximum allowable principal stress 90 MPa both in tension and in compression, and the maximum thickness of the skin panels 15 mm. The minimum skin thickness is set equal to 0.8 mm.

E.2 Initial model generation

In this example the finite element model of the fuselage to be analysed represent 90 degrees of the fuselage. Three undisturbed frames and one interrupted frame will be taken into the model. An extra offset in axial direction is defined at the start of the fuselage model. The data will be entered using a data file as described in section 3.1 and appendix A. The file used is printed here below:

```
1400. * Radius fuselage (Place your comment after the tenth column)
$
$ Comment cards may be inserted anywhere, as long as they
$ start with a $(dollar sign).
$
$ Frame input
$
4     * Number of frames
0.    * Position frame 1
500.  * Position frame 2
1000. * Position frame 3
1500. * Position frame 4
$
$ Stiffener input
$
```

Appendix E: Illustrative example

```
15    * Number of stiffeners
 0.   * Stiffener 1 offset from starting_angle_fuselage
150.  * Stiffener 2 offset from previous stringer
150.  * 3
150.  * 4
150.  * 5
150.  * 6
150.  * 7
150.  * 8
150.  * 9
150.  * 10
150.  * 11
150.  * 12
150.  * 13
150.  * 14
150.  * Stiffener 15 offset from previous stringer
$
$ cut-out size
$
650.  * height of half of the cut-out
0.    * cut-out position with respect to the datum line
400.  * width of half of the cut-out
100.  * radius of corner of the cut-out
$
$ Limit size of the structure to be analysed to the part of the
$ structure which is enclosed in the two following angles
$
0.    * starting angle of fuselage (degrees)
90.   * ending angle of fuselage (degrees)
$
$ Starting and ending offsets of the fuselage, i.e. the piece of
$ skin before the starting frame and the piece of skin after the
$ last frame (nearest to the cut-out).
$
250.  * starting offset of fuselage (mm)
 0.   * ending offset of fuselage (mm)
$
$ End of file (this card is not mandatory)
$
```

As stated previously, and also seen in the file above, comment may be inserted after the tenth column; a complete comment line must be indicated with a dollar sign in the first column. All geometrical properties mentioned earlier are found in this file. The file is read by clicking the option "Generate Fuselage" in the "Initial Geometry Menu", see figure 9, page 17.

If a model already exists, Cufus asks to remove it. When the file name is entered the initial model will be generated, and after approximately 5 minutes the model shown in figure E1 will appear on the screen.

During the generation of the model some messages will be displayed in the alpha area. These messages are intended primarily to give the user an indication of the activity of the model generator. When the model is nearly finished a data base scan is performed, see section 4.2. This scan is used to find all adjacent patches and lines for each patch. These data are used for the automatic mesh adaption. Since this data base scan is a somewhat

Appendix E: Illustrative example

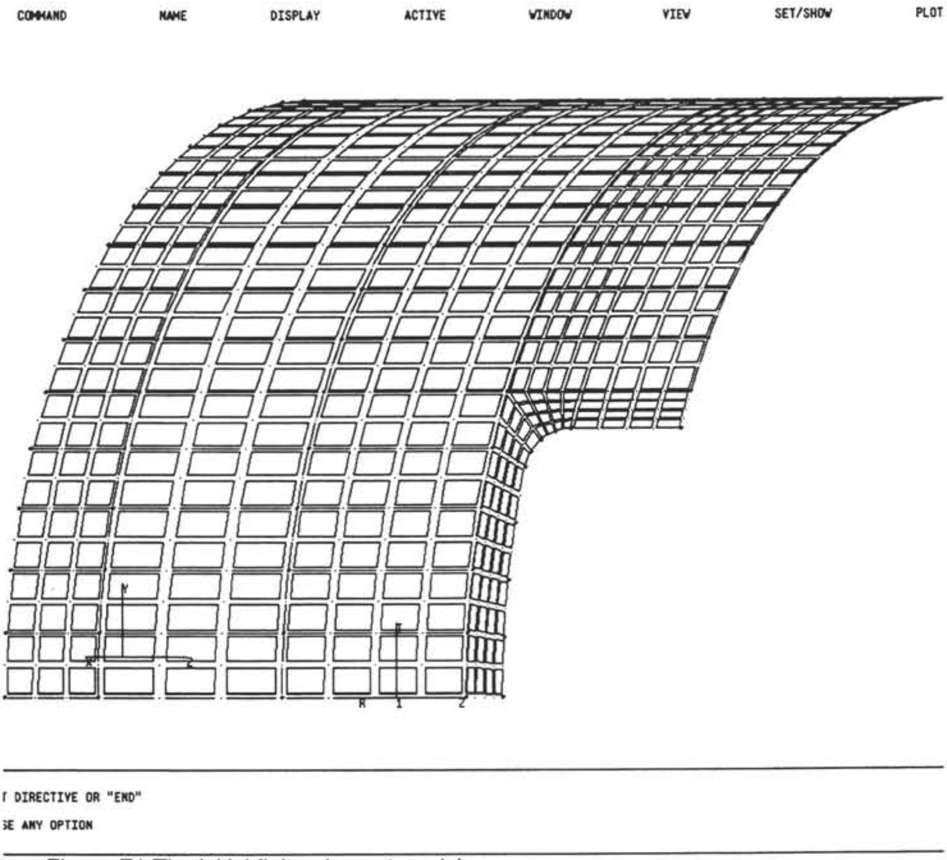


Figure E1:The initial finite element model

tedious activity for the user, the scan is visualized using different colours for the patches and lines involved.

After the initial generation of the finite element model, some PATRAN settings are changed. These settings are for display purposes only, so they do not affect the finite element model.

E.3 Geometry editing

To demonstrate the geometry editing capability, some of the cut-out dimensions will be changed. The use of this item described in section 3.4. For demonstration purposes only, the radius of the corner of the cut-out will be changed from 100 to 200 mm. The geometry editing can be invoked as showed in figure E2. As stated in section 3.4, the option for the

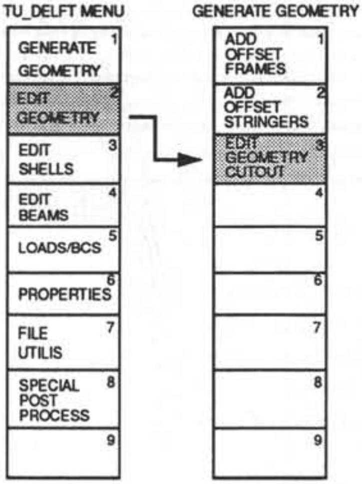


Figure E2: Starting geometry editing.

geometry editing is the only option currently implemented.

If geometry editing is activated, Cufus will prompt for three dimensions, namely the height[1] of the cut-out, the width of the cut-out and the radius of the corner of the cut-out. For all dimensions the current value is displayed. In addition the upper and lower bounds for these dimensions are also given. These limits on the variables are the result of the partial regeneration of the finite element model. The reasons for these bounds are more extensively described in section 3.4.

Values of the new height and width of the cut-out are set to the old value, i.e. they remain unchanged. For the radius of the corner of the cut-out the value 200 is entered. This will result in a partial regeneration of the finite element model, as shown in figure E3.

1. The height and the width of the cut-out used are in fact the half width and the half height of the actual cut-out, because only a quarter of the cut-out is modelled. For further details see section 3.2 (page 18).

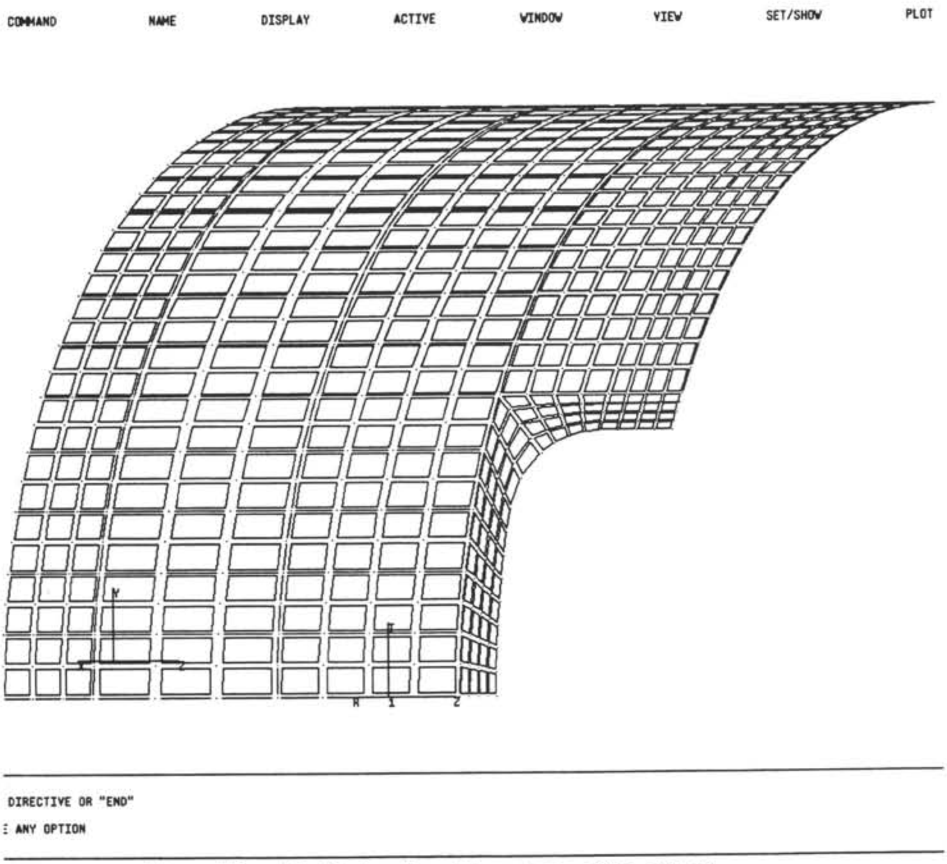


Figure E3: Modified radius of the corner of the cut-out.

The mesh densities are kept the same as before the change in the geometry. However, due to the changes in the geometry model some patches need a minor mesh adaption. Properties already assigned to the part of the finite element model to be regenerated are lost. This implies that properties of that part of the model must be assigned again.

In general it can be stated that this function should be used carefully, because the element shape can become distorted due to the partial change of geometry. To prevent a distorted element mesh, the rest of the structure should be modified as well.

If the finite element model had already been equivalenced (see section 4.3.1), it must be equivalenced again after the geometry modification. This is necessary because the geometry modification results in a partial regeneration of the finite element model, which implies that a part of the model is mutually disconnected.

For the rest of the example treated in this appendix, the radius of the corner of the cut-out will be reset to the initial 100 mm.

E.4 Element Editing

How to start element editing is described in section 4.2. This function is used to adapt the mesh in an easy way. The user can point to a patch to modify the mesh; Cufus will take care of the connectivity of the elements. Starting from the initial mesh (figure E1), the mesh usually has to be refined. If element editing has started as shown in figure 12, page 25, the result may look like figures E4 and E5.

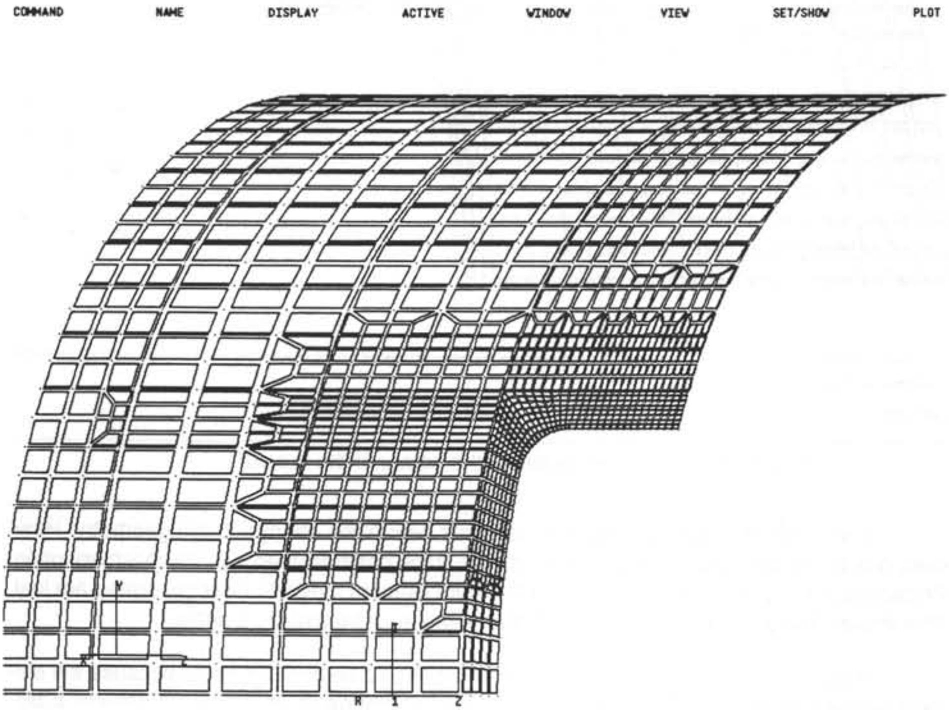


Figure E4: The finite element model after mesh adaption.

As can be seen in these figures, the program automatically defines transition regions in the mesh, to ensure a proper connectivity.

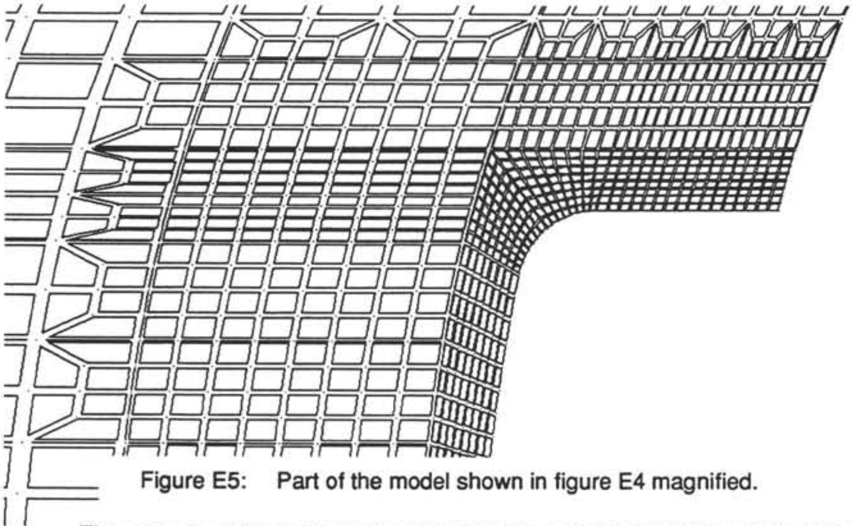


Figure E5: Part of the model shown in figure E4 magnified.

The seventh option in figure 12, see page 25, called "Delete Beams" is used to delete obsolete beams. During the mesh regeneration caused by the element editing, beams are also generated in the interior of the patches. These beams must be deleted because the only beams necessary are positioned on the boundaries of the patch. Clicking the previously mentioned option will cause a data base scan to remove all beam elements which are not positioned on a frame or stringer location. This deletion of obsolete beams is described in more detail in section 4.2.1.

E.5 Applying Boundary Conditions

Before the setting of boundary conditions is begun, it is strongly recommended first to perform node equivalencing. The node equivalencing offered by Cufus is exactly the same as in the standard PATRAN, with the exception that the Cufus node equivalence option will return automatically to the menu from which it was started. How to invoke node equivalencing is described in section 4.3.1.

In this section it will be demonstrated how to set the boundary conditions for a differential pressure.The boundary conditions can be set interactively. They could be set for a specific load case, or for all pre-defined load cases at one time.

Setting of the boundary conditions can be invoked as shown in figure 16, see page 30. When this option is clicked, a data base scan is performed for all patches. If a patch positioned at one of the boundaries of the finite element model is found, the proper boundary conditions are applied. These boundary conditions are indicated graphically by a cyan coloured circle with a few numbers. The numbers refer to the retained degrees of freedom. Sometimes the retained degrees of freedom are indicated as a series using a capital "T". For example, 1T6 means that all six degrees of freedom are retained. If a number is appended which is preceded by a capital "F", the boundary conditions are defined in a local coordinate system. For example, 246F1 means that the degrees of freedom 2, 4 and 6 in local coordinate system 1 are retained.

When the boundary conditions are set the display would look like figure E6.

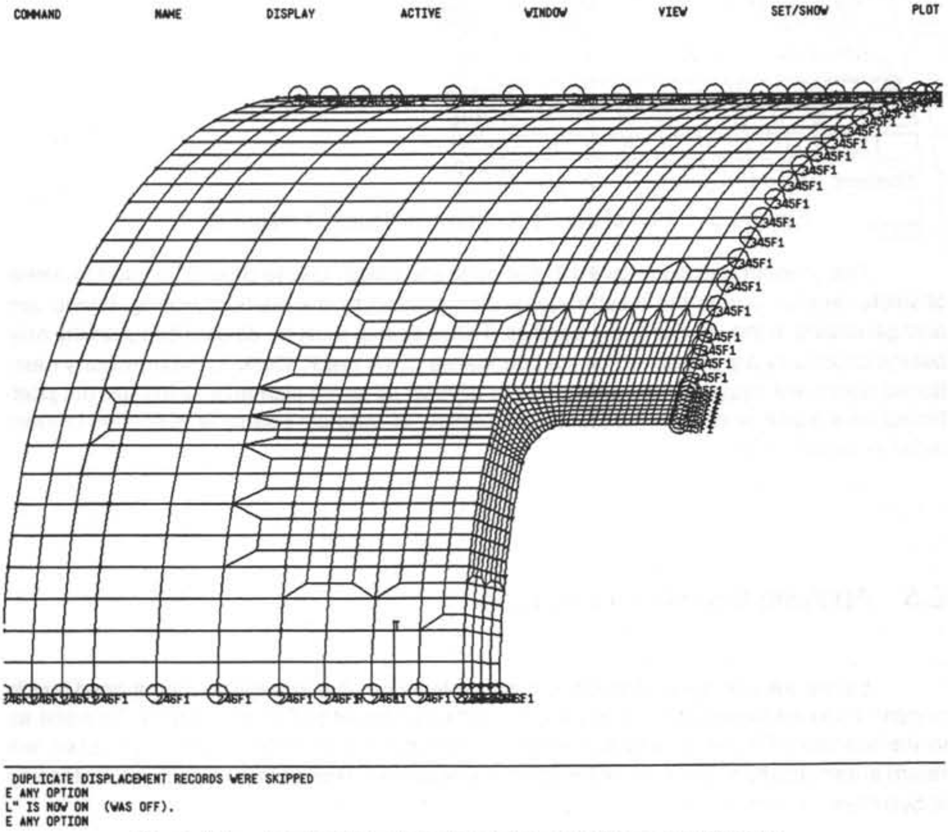


Figure E6: Application of the boundary conditions for differential pressure.

If boundary conditions for multiple load cases are set, it is easier to use the option "All BCS" of figure 16, see page 30. In this case all boundary conditions are set, using different set identification numbers. Cufus takes care of selecting the proper boundary set identification number together with the matching load set identification number.

If boundary conditions have already been defined with one of the reserved set identification numbers (see table 1, see page 31), they will be overwritten with the new boundary conditions *without* any notification.

Display of the boundary conditions will be turned off after generation. They can be made visible again by using the "model verification" option of the standard PATRAN menu [9].

E.6 Applying Loads

In this section it will be demonstrated how to apply the loads for the case of differential pressure. The loads can be applied interactively, by reading a file or by a combination of both. Here the interactive method will be used. Application of loads by reading a file is described in section 4.3.2 and appendix C.

If the option "INTERNAL PRESSURE" in figure 17 (see page 31) is selected, the user will be prompted for certain replies. First the magnitude of the differential pressure must be entered, followed by the identification number for the load case. Cufus will then apply the MPC's (multi point constraints). These MPC's are drawn as purple circles for the slave nodes connected with a purple line to the master node. These MPC's are used for a simple load introduction at the loaded side of the fuselage shell, see figure E7.

Figure E7 also shows that after the prompts for the magnitude of the differential pressure and its load case identification number, some further interactive input is required. In the figure the last two lines in the alpha area are:" Please indicate loaded node with the cursor. If you are finished, hit option 9 of the menu on the right of the screen.". This option is meant for applying loads on the structure which arise from the pressure loads on the door. The door is normally placed on so-called door stops. The loads due to the differential pressure on the door are transferred by these door stops to the fuselage. The door stops are not modelled explicitly, but the loads can be applied at this stage, by clicking one or more nodes. Each time a node is clicked the design system will ask for a percentage of the total load to be transferred by that node. The total load is calculated for the complete door and, by giving a percentage, can be distributed in an appropriate way over the structure. The total load is calculated for a quarter of the door (because only a quarter of the door is modelled in this example), so the total of individual percentages for each node should be 100 per cent.

The distribution of load on the door depends in reality on the stiffness distribution over the door and the adjustment of the door stops. Because no modelling of the door is performed and the adjustment of the door stops is difficult to control by the designer, the design system leaves the problem of a proper load distribution to the designer.

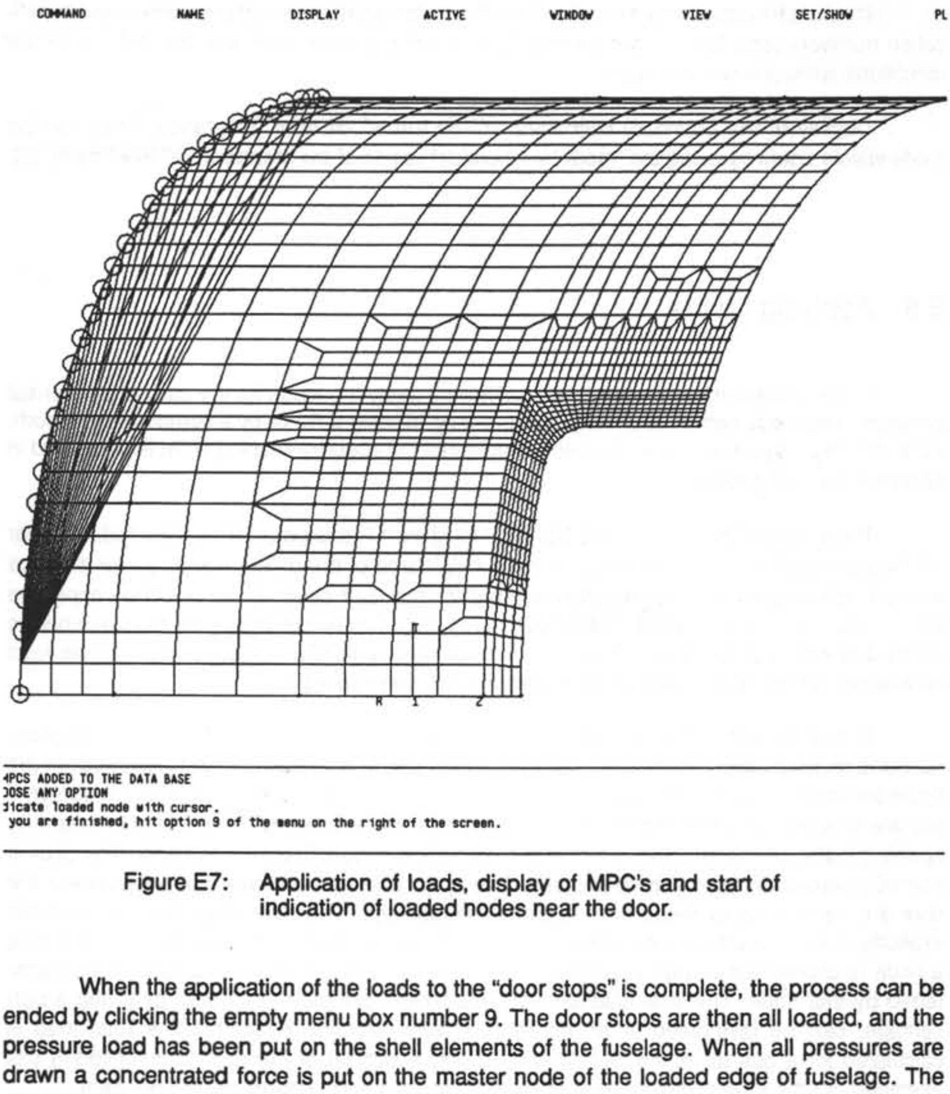


Figure E7: Application of loads, display of MPC's and start of indication of loaded nodes near the door.

When the application of the loads to the "door stops" is complete, the process can be ended by clicking the empty menu box number 9. The door stops are then all loaded, and the pressure load has been put on the shell elements of the fuselage. When all pressures are drawn a concentrated force is put on the master node of the loaded edge of fuselage. The display will be as in figure E8.

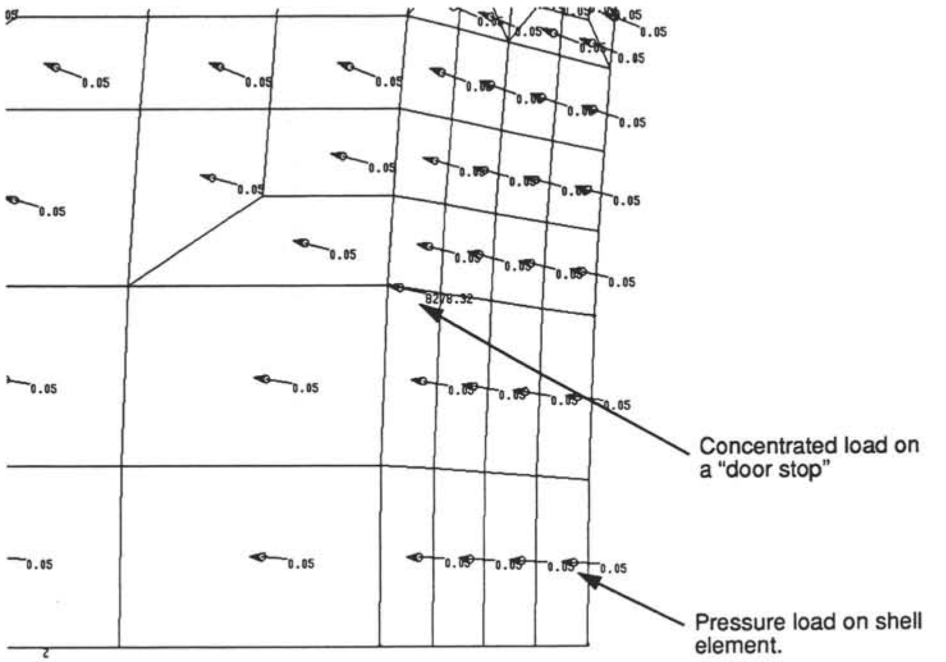


Figure E8: Application of loads, display of pressure loads and concentrated load on a door stop.

E.7 Property Definition

The definition of properties can be accomplished in several different ways, see section 4.4. One of the methods described in that section is reading a property file. The format of such a file is described in appendix C; here an example is given:

```
$
$ This file demonstrates the use of a property library together
$ with Cufus and PATRAN.
$
$ Comments may be inserted anywhere, as long as the line is preceded
$ by a "$" dollar-sign. Property records in this file are just like
```

```
the
$ data-records in the PROP-command inside PATRAN, with one exception.
$ The first ten columns are reserved for the property id.
$
$ FORMAT OF MATERIAL PROPERTIES (Same as PATRAN)
$ PMAT, mat-id, option, etc.....
$
$ FORMAT OF QUAD PROPERTIES (description MSC-NASTRAN Users manual)
$ QUAD, MID1/ T/ MID2/ (12I/T3)/ MID3/ (TS/T)/ NSM/ (Z1,Z2)/ MID4
$
$ FORMAT OF BAR PROPERTIES (description MSC-NASTRAN Users manual)
$ BAR, MID1/ A(A)/ I1(A)/ I2(A)/ I12(A)/ J(A)/ NSM(A)/ (Ci(A), Di(A),
$ Ei(A), Fi(A))/ SO/ (X/XB)/ (A, I1, I2, I12, J, NSM)/ (Ci, Di, Ei,
$ Fi)/ (K1, K2)/ (S1, S2) etc.........
$
$ Definition of MATERIALS
$ Aluminium 2024-T3
PMAT, 1, ISO, 72000., 27067.67, 0.33, 2.7E-6
$
$ Definition of QUAD-properties
$234567890
1001      QUAD,1/0.1/1
1002      QUAD,1/0.2/1
1003      QUAD,1/0.3/1
1004      QUAD,1/0.4/1
1005      QUAD,1/0.5/1
1006      QUAD,1/0.6/1
1007      QUAD,1/0.7/1
1008      QUAD,1/0.8/1
1009      QUAD,1/0.9/1
1010      QUAD,1/1.0/1
1011      QUAD,1/1.1/1
1012      QUAD,1/1.2/1
1013      QUAD,1/1.3/1
1014      QUAD,1/1.4/1
1015      QUAD,1/1.5/1
1016      QUAD,1/1.6/1
1017      QUAD,1/1.7/1
1018      QUAD,1/1.8/1
1019      QUAD,1/1.9/1
1020      QUAD,1/2.0/1
1021      QUAD,1/2.1/1
1022      QUAD,1/2.2/1
1023      QUAD,1/2.3/1
1024      QUAD,1/2.4/1
1025      QUAD,1/2.5/1
1026      QUAD,1/2.6/1
1027      QUAD,1/2.7/1
1028      QUAD,1/2.8/1
1029      QUAD,1/2.9/1
1030      QUAD,1/3.0/1
1032      QUAD,1/3.2/1
1034      QUAD,1/3.4/1
1038      QUAD,1/3.8/1
1042      QUAD,1/4.2/1
$
$ Definition of BAR-properties
```

```
$234567890
2001      BAR,1/50.86396/5460.881/941.1763/15.48465/ / / / / / / / /
/ /.4141139/.3297496
2002      BAR,1/291.7489/459066.9/11390.52/428.4573/ / / / / / / / /
/.6577195/.13035016
2003      BAR,1/866.9559/821316.9/171280.9/10039.03/ / / / / / / / /
/.4007760/.25440926
$
$
$ FORMAT OF BAR PROPERTIES (description MSC-NASTRAN Users manual)
$ BAR, MID1/ A(A)/ I1(A)/ I2(A)/ I12(A)/ J(A)/ NSM(A)/ (Ci(A), Di(A),
$ Ei(A), Fi(A))/ SO/ (X/XB)/ (A, I1, I2, I12, J, NSM)/ (Ci, Di, Ei,
$ Fi)/ (K1, K2)/ (S1, S2) etc.........
$
2301 BAR,1/63.2/4040./2350./// / / / / / / / / / / / / /
$
$ End of file (this card is not mandatory)
```

Figure E9 shows how reading the property file can be started form the menu. Apart

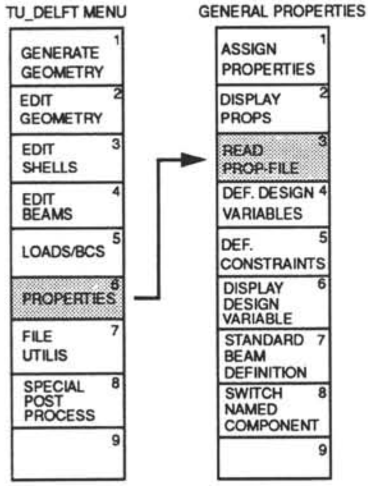


Figure E9: Start reading the property definition file.

from the addition of the properties to the standard PATRAN data base, there is no indication of whether the properties are defined or not. However, the properties in the standard PATRAN data base can be examined with the "PROP, property id, SHOW" command [9].

The other possibility for property definition referred to in section 4.4 is to use the standard beam cross-sections. In figure 21 of that section it is shown how to start this property definition. Figures 22 and 23 (see page 39), give an example of how to proceed. Finally, table 2 lists the implemented beam cross-sections. Since the example is already given in section 4.4 it is not considered useful to repeat the same example in this appendix.

In order to be able to reproduce the results given in this paragraph, the properties used are listed here. As well as reading the property file as listedon page 105, some standard beams are also defined. The dimensions used are listed in table E1.

Property identification number	name	dimensions[1] [mm]				
		h	b	u	r	t
3001	FoN 1.6308	25	20	6.0	2.5	1.0
3002	FoN 1.6302	100	50	20	15	5.0
3010	FoN 1.6302	50	15	6.0	2.5	1.0
3011	FoN 1.6302	100	40	15	10	3.6

Table E1: Dimensions of the beam sections used for the finite element model in this appendix.

1. Dimensions refer to figure 23, see page 39.

E.8 Property Assignment

The normal continuation of the property definition is the property assignment. This means that certain elements will be coupled to certain property identifications. There are numerous possibilities to assign properties to elements; these are listed in table 3, of section 4.4.3 (page 44).

Because the property assignment is discussed thoroughly in section 4.4.2, it is not considered useful to repeat large parts of that section here. Therefore, it is sufficient to show only the results of the property assignment here. Since the quality of grey-scale plots on a black and white hardcopy device is rather poor, the plots are reproduced here using hatching instead, see figures E10a and E10b.The bold numbers used in these figures refer to the property identification numbers of the previous section.

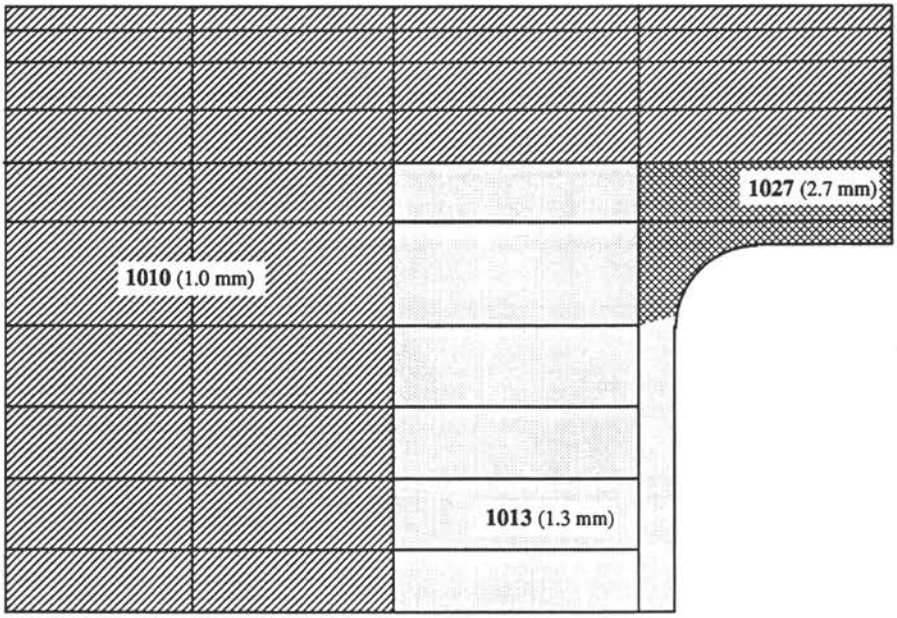


Figure E10a: The used shell properties. The bold numbers refer to the property identification numbers, see section E.7.

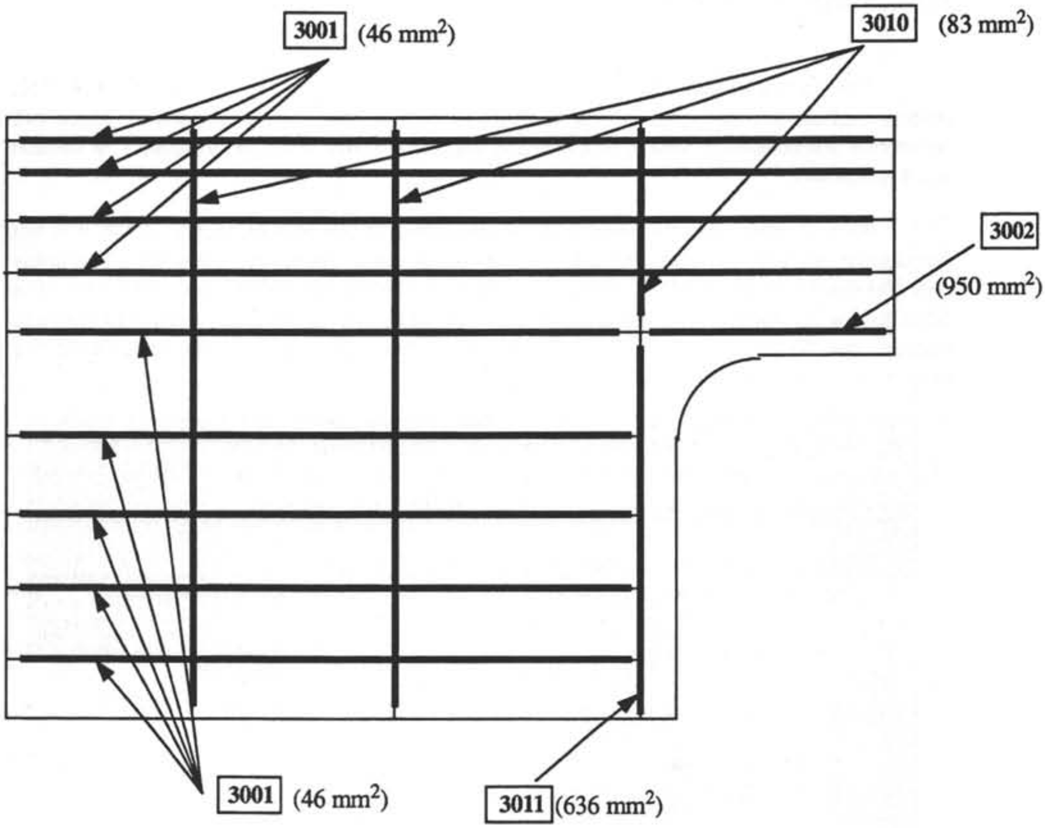


Figure E10b: The beam properties used. The bold numbers refer to the property identification numbers, see section E.7.

E.9 Preparation of MSC/NASTRAN analysis

The finite element model is now defined completely for a static analysis. In this section it is shown how the PATRAN model can be translated to a MSC/NASTRAN model. Also translation of MSC/NASTRAN results to a PATRAN format will be discussed.

The finite element model can be written to a neutral file. This file is readable for a number of different interfaces supplied with PATRAN. The file can be generated as shown in section 2.5 (page 11), and in chapter 29 of the PATRAN User's Manual [9]. As well as the neutral file a Case Control Deck can also be generated automatically, see section 4.3.3. How to generate this Case Control Deck is shown in figure 19 of that section.

When these two files are generated, the PATRAN window can be closed. The interface is started by typing "patnas" at the command prompt[1]. The use of PATNAS is described in detail in PAT/MSC-NASTRAN application interface [10]. One remark has to be made with the use of this interface. During the input phase of the interface the name of the Case Control Deck is required. This is the file which is generated by Cufus. If the name of the Case Control Deck generated by Cufus is entered the interface will merge both files.

When the interface has completed its task, some minor manual editing can take place. For a linear static analysis there are two lines in the Case Control Deck which can be moved to the Bulk Data Deck. It is not mandatory to make these changes, but it is recommended. For a non-linear static analysis a NLPARM-card has to included in the Bulk Data Deck [11]. This card controls the non-linear solver of MSC/NASTRAN. Because the values used on this card are not "generic" values, this card is not generated automatically by Cufus.

The file is now ready. A job file also has to be written, section A.4. When the job file is written, MSC/NASTRAN can be executed.

NASTRAN returns the results in a binary file called "fort.12". This file must be translated with NASPAT [10] into result files which are readable by PATRAN. Displaying the contents of these files is covered by the PATRAN User's Manual [9]. Therefore it is considered sufficient to show only the results in this appendix. Figure E11a shows the stresses in column 12 (major principal stress in the first side of the shell) for a linear static calculation, while figure E11b shows the same stresses for the same structure and loading for a geometrically non-linear calculation.

The differences in the results showed by these figures is rather limited. The peak stresses differ slightly for the geometrically linear and non-linear static calculation. This confirms the hypothesis stated in section 2.2, in which it was stated that geometrically non-linear effects are very limited at stress levels normally occurring in pressurized fuselages (i.e. < 100 MPa).

1. For working at the T.U. Delft, this command can only be run at dutlfs4, see appendix A.

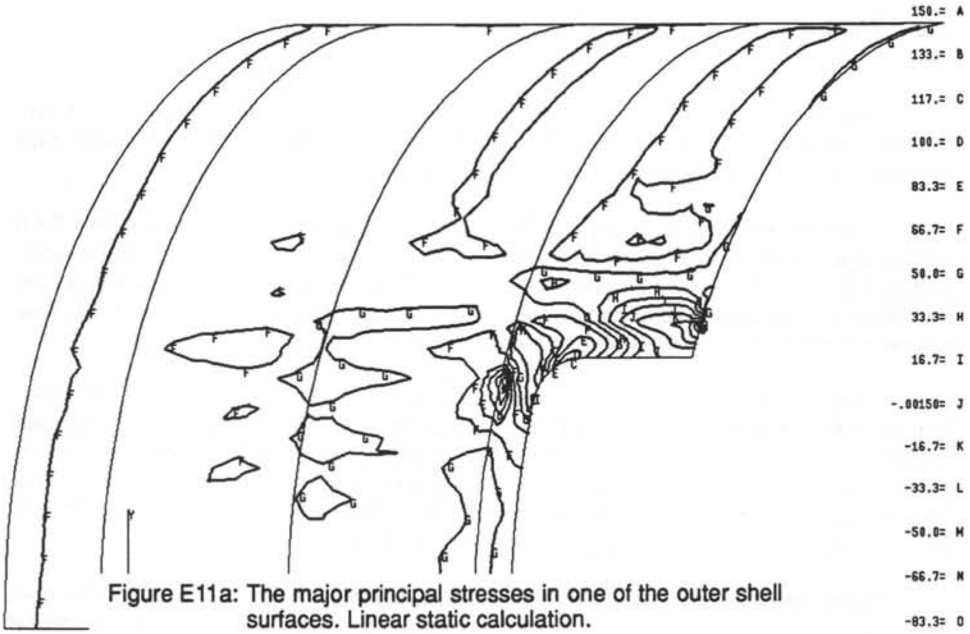


Figure E11a: The major principal stresses in one of the outer shell surfaces. Linear static calculation.

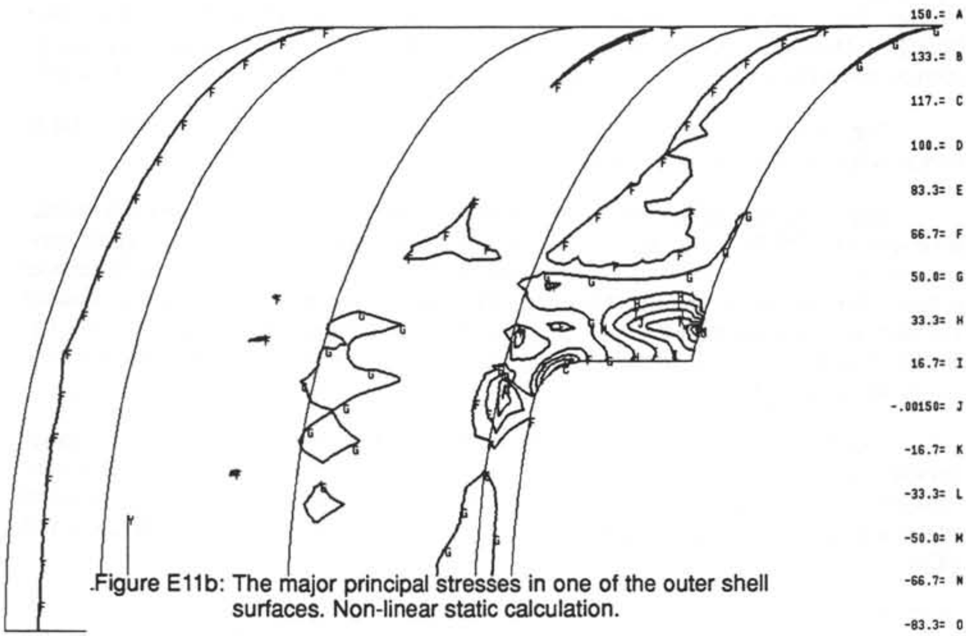
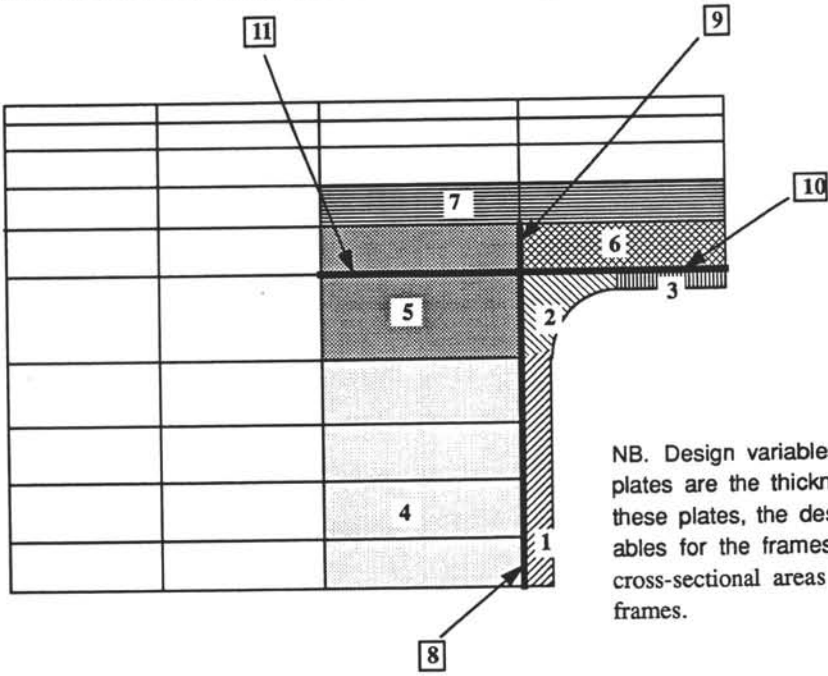


Figure E11b: The major principal stresses in one of the outer shell surfaces. Non-linear static calculation.

E.10 Definition of Design Variables and Constraints

The definition of design variables was discussed thoroughly in section 4.4.3 and the definition of constraints in section 4.4.4. Because these sections are regarded as sufficient to explain the definition of design variables and constraints, only the results used for the example will be listed in this section. Figure E12 shows the design variables used for both the shell



NB. Design variables for the plates are the thicknesses of these plates, the design variables for the frames are the cross-sectional areas of these frames.

Figure E12: The used design variables for the plates and frames.

elements and the beam elements. The design variables for the shell elements are the thicknesses and for the beam elements the cross-sectional areas.

The constraints used in the example are a maximum allowable principal stress of 90 MPa, both in tension and in compression in the shell elements. These constraints act on the complete finite element model.

E.11 Pre-processing Sensitivity Analysis

The main part of the pre-processing of a sensitivity analysis (and optimization) is described in the previous section, although there is some specific input data required by MSC/NASTRAN for both analysis types. These data consist of the Case Control Deck as well as an extension to the Bulk Data Deck of the MSC/NASTRAN input file. These extra data can be generated by clicking the option "Sensitivity CCD", as shown in figure E13. In the a similar

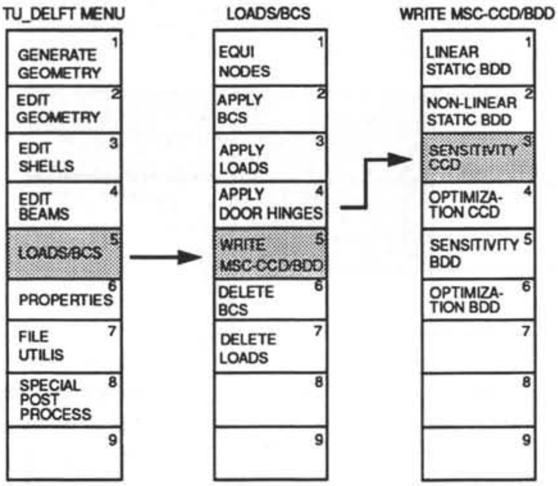


Figure E13: Generation of the MSC/NASTRAN Case Control Deck for an sensitivity analysis.

way an extension to the MSC/NASTRAN Bulk Data Deck can be generated. When option 5, "Sensitivity BDD" is chosen an extension to the Bulk Data Deck is written.

When these extensions are ready, the NASTRAN job is generated in the normal way, which implies a neutral file is generated [7] and translated by the interface program PATNAS [10]. During the execution of PATNAS the Case Control Deck generated by Cufus can be included. The extension to the Bulk Data Deck, however, must be done by hand, using a text editor, after the execution of PATNAS is completed.

Depending on the computing environment a job-file (see appendix A) completes the pre-processing of the sensitivity analysis, and the MSC/NASTRAN job can be submitted.

E.12 Post-processing Sensitivity Analysis

The post-processing of a sensitivity analysis is described in detail in section 5.1. The post-processing is prepared by choosing menu option "Generate sens files", figure 34, see page 57. This option reads a so-called MSC/NASTRAN punch file. This file is sorted in such a way that it can be used by Cufus. The files generated by this option can be used by choosing the second option "Display default sens", figure 34. The option "User defined sens" does work, but needs different files to the default sensitivity files. These files cannot yet be generated automatically. This option is intended for displaying different normalizations of the sensitivity coefficient. Now the standard sensitivity coefficient is shown. That means the figures display the ***change*** in stresses due to a ***doubling*** of the value of the design variable to be changed.

If the option "DISPLAY DEFAULT SENS" is chosen the alpha area displays the available response codes. These response codes refer to the responses which were defined during the definition of the constraints. A partial list of the response codes can be found in table E2. The complete list can be found in the MSC/NASTRAN User's Manual [11]. It must be emphasized that post-processing of a sensitivity analysis can *only* be performed on the

QUAD4 element[1]	
Response code	Response
3	Normal x at Z_1
4	Normal y at Z_1
5	Shear xy at Z_1
6	Shear angle at Z_1
7	Major principal at Z_1
8	Minor principal at Z_1
9	Von Mises or maximum shear at Z_1
11	Normal x at Z_2
12	Normal y at Z_2
13	Shear xy at Z_2
14	Shear angle at Z_2
15	Major principal at Z_2
16	Minor principal at Z_1
17	Von Mises or maximum shear at Z_2

Table E2: The currently used response codes for Sensititvity analysis.

1. Z_1 and Z_2 refer to the outer sides of the shell elements by default.

responses requested at the constraint definition during the pre-processing phase. For the example problem used in this appendix the major principal stresses on both sides of the shell were constrained. In the example sensitivity file for code 15 (major principal stress on side 2 of the element) is chosen. Then the program asks to enter one of the numbers of the design variables. When design variable 7 is chosen a picture like figure E14 will appear on the screen.

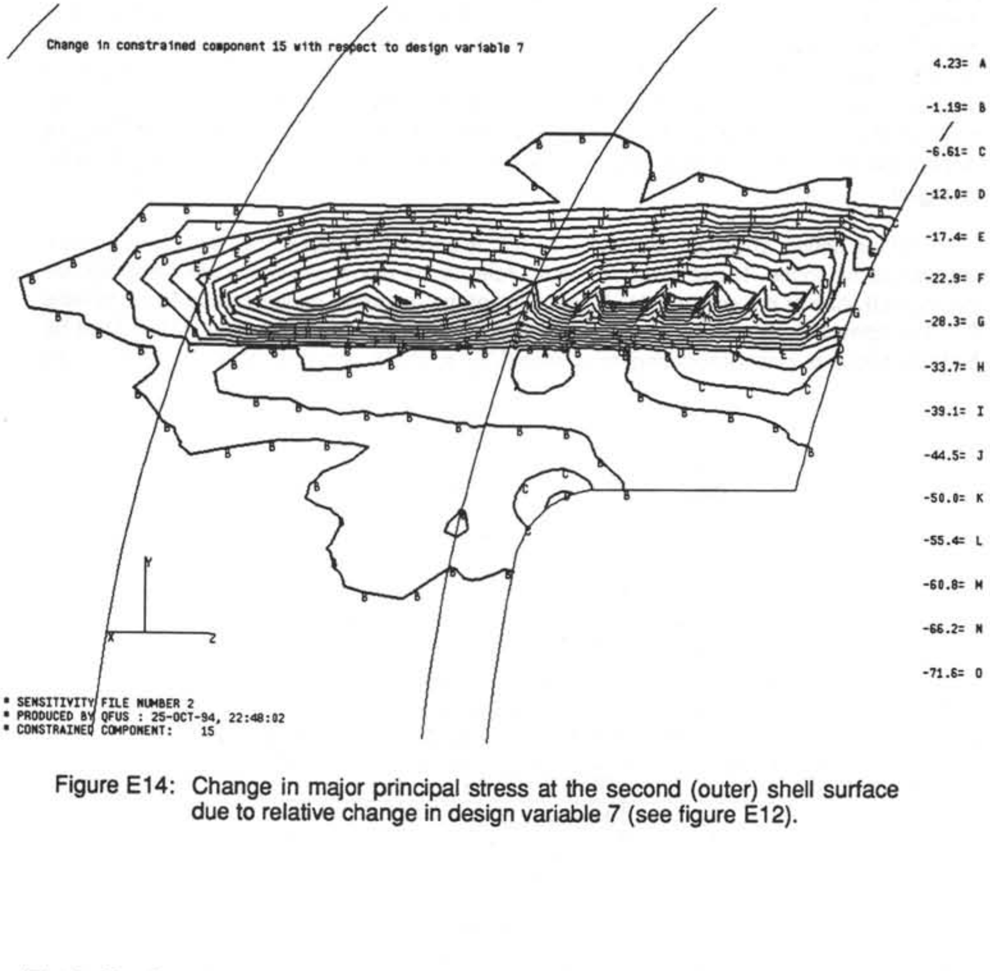


Figure E14: Change in major principal stress at the second (outer) shell surface due to relative change in design variable 7 (see figure E12).

E.13 Performing a "What-if" analysis

The "What-if" analysis is a special way of post-processing a sensitivity analysis. Therefore the preparations for a "What-if" analysis are similar to the post-processing of a sensitivity analysis. The sensitivity files, see section E.12, must be available before the actual "What-if" analysis can take place. Obviously, if the sensitivity files are already generated for the post-processing of a sensitivity analysis, they do not need to be generated again. If the files are available the menu option "Prepare What-If", figure 34, see page 57 is chosen. This option results in the loading of the sensitivity data and the stresses into the core memory of the computer. The program will ask for the file containing the element stresses. ***The format of the element stress file must be ASCII format.*** The interface program NASPAT [10] offers the possibility to write the output files as well in binary format or text (ASCII) format. The default format is the binary format. If the file is written to ASCII format, the file containing the ASCCI data does not overwrite the original file. Normally the results are written to a file called "text.els". Obviously this file has to be read by Cufus. Even if the file is written in binary format it can be converted into text format by executing an utility program called "READER". This program is a part of the PATRAN software package [7]. Using this option is necessary only once per session and finalizes the preparation of the "What-if" analysis. This operation typically takes 5 to 10 minutes.

Now the actual "What-if" analysis can take place. In the example used in this appendix the value of design variable 1 (figure E12) will be changed from 1.3 to 1.6 mm and subsequently the value of design variable 2 will changed from 2.7 to 3.0 mm. The stress distribution (major principal stress at side 1 of the shell elements) is shown in figure E15.

E.14 Pre-processing Optimization

This section can be very short. The specific optimization pre-processing is similar to the specific sensitivity analysis pre-processing described in section E.11. The only (obvious) exceptions are that instead of options 3 and 5 in figure E9, the options 4 "OPTIMIZATION BDD" and 6 "Optimization CCD" are chosen respectively, during the generation of the Bulk Data Deck the value of the finite difference step must be entered. A default value is suggested by the program and will in most cases suffice, see also the discussion in section 4.3.3 (page 32). The rest of the pre-processing of the optimization is exactly the same as the pre-processing of the sensitivity analysis.

E.15 Post-processing Optimization

In the section dealing with the post-processing options offered by Cufus, see section 5.2, it is stated that these options consist of the generation of 2D graphs. These graphs show the most important optimization history data. The data shown in these graphs are the objec-

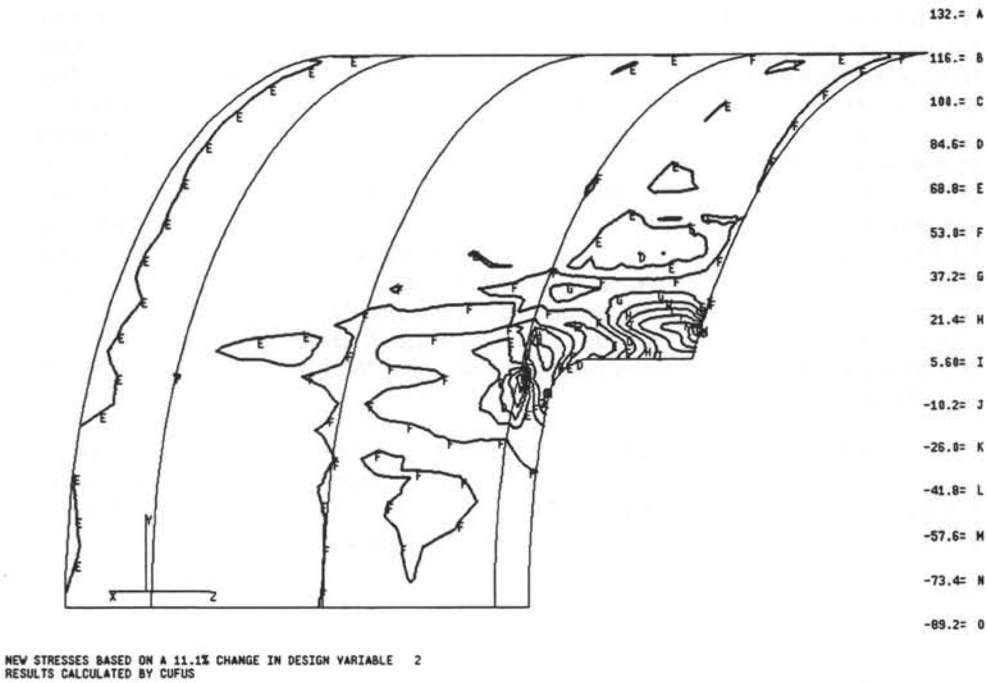


Figure E15: Stress distribution predicted by a "What-if" analysis.

tive function, the design variables and the maximum constraint violation, all against the number of design cycles. The data required for the graphs must be delivered in a so-called xyd-file. This file types can be read by P/PLOT, a separate program, part of the PATRAN software package. The format of these files is described in PATRAN Plus User Manual [7]. However, the user of Cufus need not be bothered by the format of these files. Along with Cufus a FORTRAN 77 program called "scan68.f" is included[1]. This program reads the output file (f06-file) of MSC/NASTRAN, and generates a xyd-file. Assuming that the FORTRAN program scan68.f is compiled to an executable program called "sc68" the UNIX command to generate the xyd-file will be:

```
sc68 < job-name.f06 > job-name.xyd ↵
```

1. The program scan68.f is meant for scanning a MSC/NASTRAN version 68 output file. There are also programs called scan675.f and scan67R2.f, which operate similar to scan68.f, but works on MSC/NASTRAN version 67.5 and 67 Release 2 output files respectively.

In this command line `job-name.f06` stands for the MSC/NASTRAN output file and `job-name.xyd` for the xyd-file.

If the xyd-file is generated, option 4 of figure E16 "X-Y Graphs Optimization", can be chosen. This option will ask for the xyd-file. After this file name is entered a so-called tem-

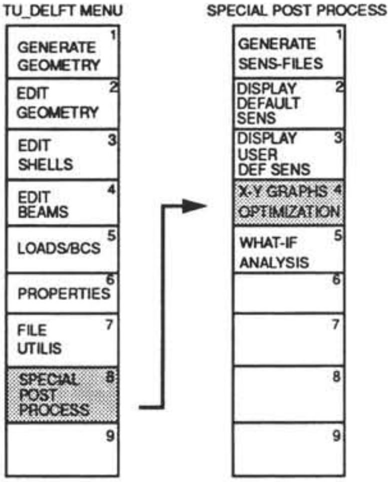


Figure E16: Starting the post-processing of the structural optimization.

plate file is generated. The template file in combination with xy-data-file are required by P/PLOT. Then P/PLOT will be started as a child process from PATRAN. The graphs will be read and displayed. Using standard P/PLOT commands allow the user to view the different graphs one by one. However, there are some utilities in Cufus offering an easy way to handle and to display the several graphs. How to use these utilities is described in section 5.2, page 56. The graphs shown in this appendix are all made using these utilities.

If the finite element model used as example in this appendix is optimized and the results post-processed, the graph shown in figure E17 can be displayed. It is a good custom to perform an optimization more than once, starting from different initial designs. Therefore two extra optimizations are performed where the initial values of the design variables are changed into more or less random values. The history of the objective functions are shown in figure E18. This figure shows clearly that the two optimizations performed from a "random" initial design reach a lower value than the original initial design. Further investigations into the behaviour of the first optimization show the large influence of the finite difference step, see section 4.3.3. The effect of this parameter is shown in figure E19. As can be seen from that figure, even for an optimum value of the finite difference step, the optimization does not reach the true optimum value of the objective function. Restarting the optimization, i.e. using the "old" optimum as initial design for a new optimization, showed an improvement in the results.

Appendix E: Illustrative example

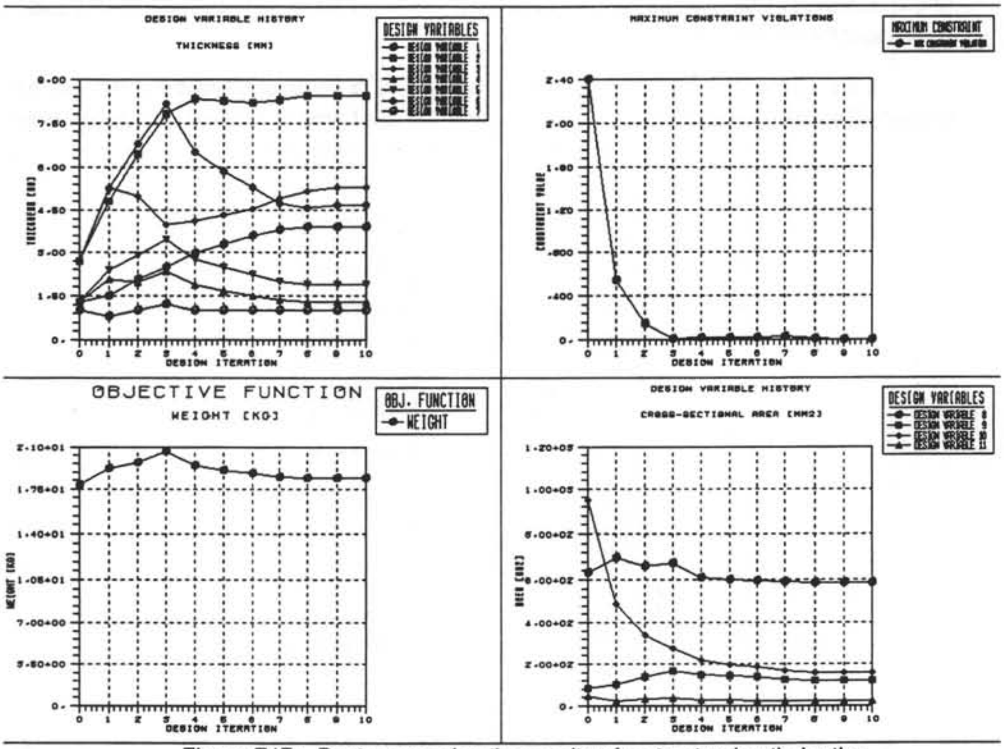


Figure E17: Post-processing the results of a structural optimization.

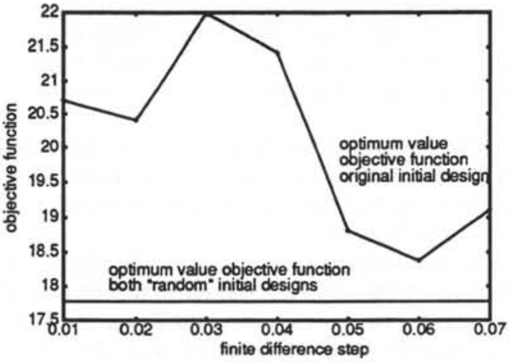


Figure E19: The effect of the finite difference step on a particular optimization.

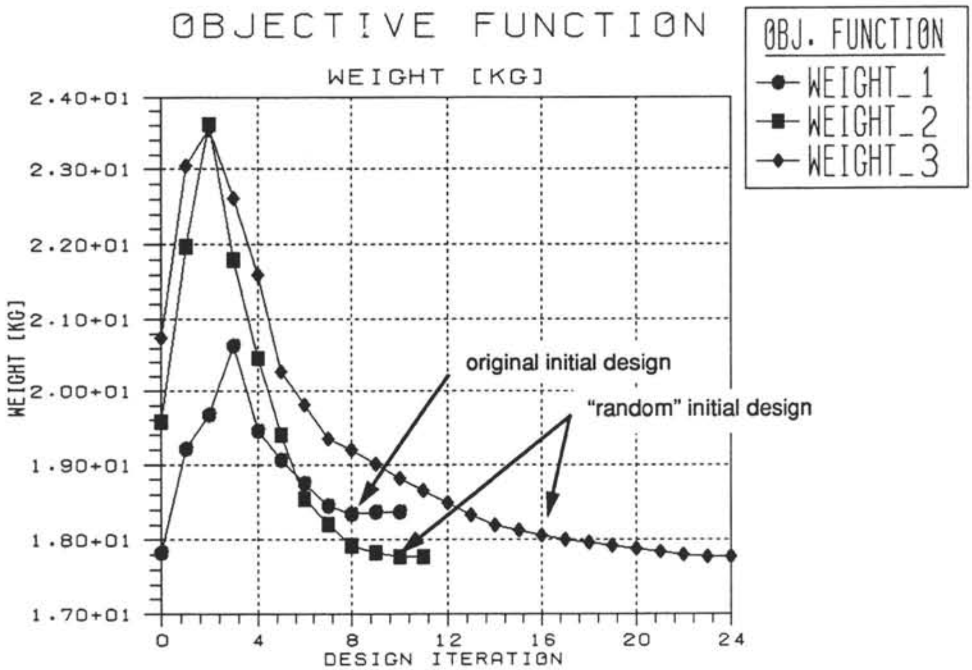


Figure E18: Objective function history of multiple structural optimizations.

However to reach the true optimum, several restarts must be performed. This could be an indication that optimization using the original initial design suffers from a "shallow part" of the design space. MSC/NASTRAN uses a gradient based optimization procedure [12]. Regions in the design space which are very shallow, i.e. the objective function or the constraints only vary slightly with a large change in the design variables, can result in a premature termination of the optimization. The fact that the finite difference step, used for the calculation of the gradients of constraints and the objective function, has a large influence on the optimization is also a clue pointing in the direction of a problem with the steepness of the gradients.

From the previous example the necessity of checking an optimization by a few extra optimizations, using different initial designs, is clearly demonstrated. Since the original initial design resulted in an unstable behaviour of the optimization, one of the "random" initial designs is chosen to define the optimum design. The initial values of the design variables and the optimum values of the design variables are listed in table E3. The complete set of postprocessing graphs of this optimization is showed in figure E20.

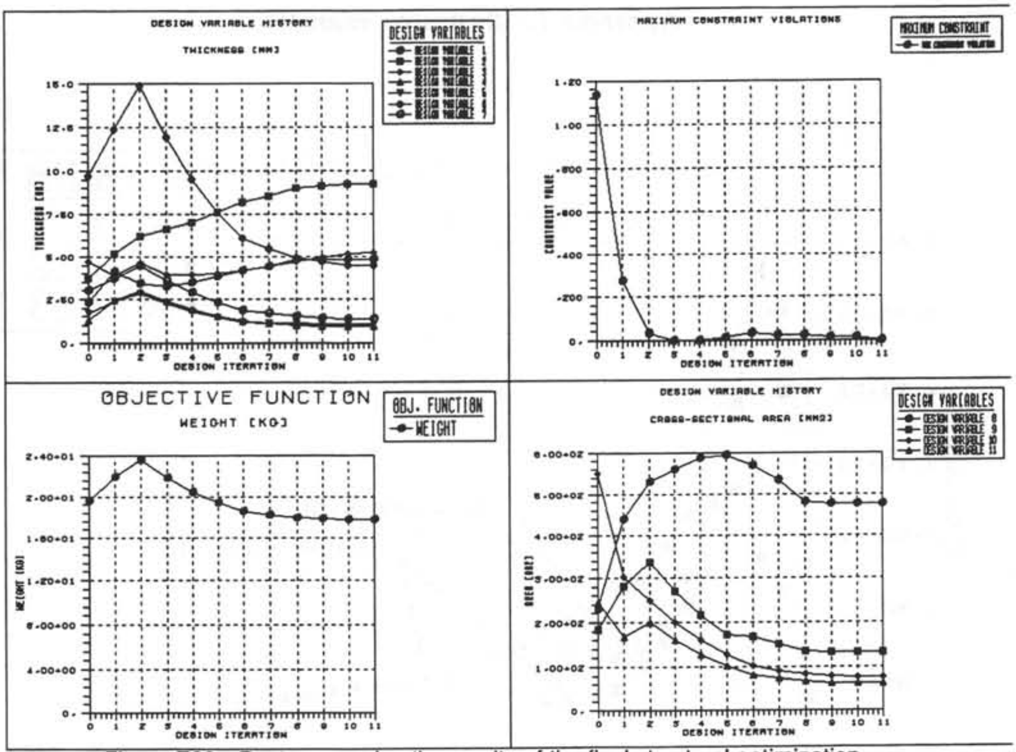


Figure E20: Post-processing the results of the final structural optimization.

design variable	initial value	optimum value
1	2.3000 mm	4.8451 mm
2	3.7000 mm	9.2327 mm
3	4.7000 mm	5.2060 mm
4	1.3000 mm	0.9424 mm
5	1.7000 mm	1.0771 mm
6	9.7000 mm	4.4824 mm
7	3.0000 mm	1.3489 mm
8	231.23 mm^2	476.12 mm^2
9	182.85 mm^2	133.20 mm^2
10	549.78 mm^2	77.263 mm^2
11	246.43 mm^2	63.220 mm^2

Table E3: The values of the design variables.

Appendix F: Quick Reference

F.1 Introduction

In this appendix no explanations for the use of Cufus functions are given. Here is only indicated how to travel through the menus to find a certain function or option. The starting point for these menu paths is always the Cufus root menu, or the PATRAN root menu. In the latter case that will be explicitly mentioned. The form of the menu path is given like in the following example:

menu 1 menu 2 menu 3 menu 4

F.2 Initial model generation

Generation of fuselage

Generate Geometry Generate Fuselage

Generation of door

Generate Geometry Generate Door

Deletion of Door

Generate Geometry Delete Door

F.3 Geometry editing

Adding offset to frames (castellations)[1]

Edit Geometry Add Offset Frames

1. Not (yet) implemented.

Adding offset to stringers[1]

Edit Geometry	Add Offset Stringers

Geometry editing of the cut-out

Edit Geometry	Edit Geometry Cut-Out

Deletion of lines at a certain frame station

Edit Geometry	Delete Station Lines

F.4 Element Editing

Patch mesh refinement in axial direction

Edit Shells	Fine Mesh Axial

Patch mesh refinement in circumferential direction

Edit Shells	Fine Mesh Circum.

Patch mesh refinement in both directions

Edit Shells	Fine Mesh Both

Patch mesh coarsening in axial direction

Edit Shells	Coarse Mesh Axial

Patch mesh coarsening in circumferential direction

Edit Shells	Coarse Mesh Circum.

Patch mesh coarsening in both directions

Edit Shells	Coarse Mesh Both

Line mesh refinement

Edit Beams	Fine Line

Line mesh coarsening

Edit Beams	Coarse Line

Deletion of obsolete beam elements

Edit Shells	Delete Beams

or

Edit Beams	Delete Beams

1. Not (yet) implemented.

F.5 Applying Boundary Conditions

Application of boundary conditions for differential pressure and axial force

Loads/BCS	Apply BCS	p & Nz BCS

Application of boundary conditions for N_y and M_x

Loads/BCS	Apply BCS	Ny & Mx BCS

Application of boundary conditions for N_x and M_y

Loads/BCS	Apply BCS	Nx & My BCS

Application of boundary conditions for M_z

Loads/BCS	Apply BCS	Mz BCS

Application of boundary conditions for all load cases

Loads/BCS	Apply BCS	All BCS

Deletion of all boundary conditions

Loads/BCS	Delete BCS

F.6 Applying Loads

Application of differential pressure

Loads/BCS	Apply Loads	p

Application of force N_x

Loads/BCS	Apply Loads	Nx

Application of force N_y

Loads/BCS	Apply Loads	Ny

Application of force N_z

Loads/BCS	Apply Loads	Nz

Application of moment M_x

Loads/BCS	Apply Loads	Mx

Application of moment M_y

Loads/BCS	Apply Loads	My

Application of moment M_z

Loads/BCS	Apply Loads	Mz

Read loads file

Loads/BCS	Apply Loads	Read File

Deletion of all loads

Loads/BCS	Delete Loads

F.7 Property Definition

Read property file

Properties	Read Prop-File

Definition of standard beam sections

General Properties	Standard Beam Def. (starts graphical menu)

F.8 Property Assignment

Assigning a property identification number for shell elements to a single patch

Properties	Assign Properties	Patch Props	Single Patch

Assigning a property identification number for shell elements to a group of patches

Properties	Assign Properties	Patch Props	Group Patches

Assigning a property identification number for shell elements to all patches

Properties	Assign Properties	Patch Props	All Patches

Assigning a property identification number for shell elements to all active patches

Properties	Assign Properties	Patch Props	Active Patches

Assigning a property identification number to a single element

Properties	Assign Properties	Element Props	Single Element

Assigning a property identification number to a group of elements

Properties	Assign Properties	Element Props	Group Elements

Assigning a property identification number to all elements

Properties	Assign Properties	Element Props	All Elements

Assigning a property identification number to all active elements

Properties	Assign Properties	Element Props	Active Elements

Assigning a property identification number for beam elements to a single line

Properties	Assign Properties	Line Props	Single Line

Assigning a property identification number for beam elements to a group of elements

Properties	Assign Properties	Line Props	Group Lines

Assigning a property identification number for beam elements to all elements

Properties	Assign Properties	Line Props	All Lines

Assigning a property identification number for beam elements to all active elements

Properties	Assign Properties	Line Props	Active Lines

Assigning a property identification number for beam elements to a single stringer

Properties	Assign Properties	Line Props	Single Stringer

Assigning a property identification number for beam elements to all stringers

Properties	Assign Properties	Line Props	All Stringers

Assigning a property identification number for beam elements to a single frame

Properties	Assign Properties	Line Props	Single frame

Assigning a property identification number for beam elements to all frames

Properties	Assign Properties	Line Props	All frames

Displaying shell properties (also occurs on several other places)

Properties	Display Props	QUAD/TRI props

Displaying bar properties (also occurs on several other places)

Properties	Display Props	BAR props

Displaying all properties (also occurs on several other places)

Properties	Display Props	All props

F.9 Preparation of MSC/NASTRAN analysis

Create a case control deck for a linear static finite element analysis

Loads/BCS	Write MSC-CCD/BDD	Linear Static CCD

Create a case control deck for a non-linear static finite element analysis

Loads/BCS	Write MSC-CCD/BDD	Non-Linear Static CCD

Create a case control deck for a sensitivity analysis

Loads/BCS	Write MSC-CCD/BDD	Sensitivity CCD

Create a case control deck for an optimization

Loads/BCS	Write MSC-CCD/BDD	Optimization CCD

Create an extension to the bulk data deck for a sensitivity analysis

Loads/BCS	Write MSC-CCD/BDD	Sensitivity BDD

Create an extension to the bulk data deck for a n optimization

Loads/BCS	Write MSC-CCD/BDD	Optimization BDD

F.10 Definition of Design Variables and Constraints

Definition of design variable for a single QUAD or TRI element

Properties	Def. Design Variable	Quad DV Selection	Select Single Quad/Tri

Definition of design variable for a group of QUAD and/or TRI elements

Properties	Def. Design Variable	Quad DV Selection	Select Group Quad/Tri's

Definition of design variable for a single patch

Properties	Def. Design Variable	Quad DV Selection	Select Single Patch

Definition of design variable for a group of patches

Properties	Def. Design Variable	Quad DV Selection	Select Group Patches

Definition of design variable for a single BAR element

Properties	Def. Design Variable	Bar DV Selection	Select Single Bar

Definition of design variable for a group of BAR elements

Properties	Def. Design Variable	Bar DV Selection	Select Group Bar's

Definition of design variable for a single line

Properties	Def. Design Variable	Bar DV Selection	Select Single Line

Definition of design variable for a group of lines

Properties	Def. Design Variable	Bar DV Selection	Select Group Lines

Displaying design variables (occurs also in many other menu's)

Properties	Display DV's

Definition of a constraint applicable to QUAD (or TRI) elements. This menu options will start some other menu's for the further definition of the constraint. Since those menu's are expected to be self-explaining, they won't be reproduced here.

Properties   Def. Constraint   Quad Constraint

Definition of a constraint applicable to BAR elements. This menu options will start some other menu's for the further definition of the constraint. Since those menu's are expected to be self-explaining, they won't be reproduced here.

Properties   Def. Constraint   Bar Constraint

Definition of a nodal constraint applicable to a single or all nodes. This menu options will start some other menu's for the further definition of the constraint. Since those menu's are expected to be self-explaining, they won't be reproduced here.

Properties   Def. Constraint   Nodal Constraint

F.11 Editing Design Variables and Constraints

Editing design variables. This menu options will start some other menu's for various editing options. Since those menu's are expected to be self-explaining, they won't be reproduced here.

Properties   Def. Design Variable   Edit Design Variable

Editing constraints. This menu options will start some other menu's for various editing options. Since those menu's are expected to be self-explaining, they won't be reproduced here.

Properties   Def. Constraint   Edit Constraint

Delete all design variables. This menu options deletes all Cufus data concerning the design variables. The PATRAN data (property identification numbers) must be edited manually.

Properties   Def. Design Variable   Del All Design Variables

Delete all constraints.

Properties   Def. Constraint   Del All Constraints

F.12 General Utilities

Switching from modelling the fuselage to the modelling of the door

Properties	Switch Named Component

Write dump file

File Utils	Save Data

Read dump file

File Utils	Restart

Deletion of all patran.nod.* files

File Utils	Remove All NOD Files

F.13 Special Post-processing Options

Preparing the post-processing of a sensitivity analysis

Special Post Process	Generate Sens Files

Displaying the results of a sensitivity analysis using the standard MSC/NASTRAN sensitivity coefficient

Special Post Process	Display Default Sens

Displaying the results of a sensitivity analysis using an user-defined normalization of the sensitivity coefficient[1]

Special Post Process	Display User Def Sens

Displaying the results of an optimization

Special Post Process	X-Y Graphs Optimization

Preparing the post-processing of a what-if analysis, step 1 (if the sensitivity files are already present this step is not needed)

Special Post Process	Generate Sens Files

Preparing the post-processing of a what-if analysis, step 2 (this step is required only once per session)

Special Post Process	What-If	Prepare What-If

What-if analysis, modifying a design variable containing shell elements

Special Post Process	What-If	What-If (Shell)

1. Not yet completely implemented.

What-if analysis, modifying a design variable containing BAR elements

Special Post Process	What-If	What-If (Bar)

What-if analysis, writing the modified values of the design variables to the actual property identification numbers

Special Post Process	What-If	Update Properties

Index

E

F

G

L

M

N

O

P

R

S

W

Series 01: Aerodynamics

01. F. Motallebi, 'Prediction of Mean Flow Data for Adiabatic 2-D Compressible Turbulent Boundary Layers'
1997 / VI + 90 pages / ISBN 90-407-1564-5
02. P.E. Skåre, 'Flow Measurements for an Afterbody in a Vertical Wind Tunnel'
1997 / XIV + 98 pages / ISBN 90-407-1565-3
03. B.W. van Oudheusden, 'Investigation of Large-Amplitude 1-DOF Rotational Galloping'
1998 / IV + 100 pages / ISBN 90-407-1566-1
04. E.M. Houtman / W.J. Bannink / B.H. Timmerman, 'Experimental and Computational Study of a Blunt Cylinder-Flare Model in High Supersonic Flow'
1998 / VIII + 40 pages / ISBN 90-407-1567-X
05. G.J.D. Zondervan, 'A Review of Propeller Modelling Techniques Based on Euler Methods'
1998 / IV + 84 pages / ISBN 90-407-1568-8
06. M.J. Tummers / D.M. Passchier, 'Spectral Analysis of Individual Realization LDA Data'
1998 / VIII + 36 pages / ISBN 90-407-1569-6
07. P.J.J. Moeleker, 'Linear Temporal Stability Analysis'
1998 / VI + 74 pages / ISBN 90-407-1570-X
08. B.W. van Oudheusden, 'Galloping Behaviour of an Aeroelastic Oscillator with Two Degrees of Freedom'
1998 / IV + 128 pages / ISBN 90-407-1571-8
09. R. Mayer, 'Orientation on Quantitative IR-thermografy in Wall-shear Stress Measurements'
1998 / XII + 108 pages / ISBN 90-407-1572-6
10. K.J.A. Westin / R.A.W.M. Henkes, 'Prediction of Bypass Transition with Differential Reynolds Stress Models'
1998 / VI + 78 pages / ISBN 90-407-1573-4
11. J.L.M. Nijholt, 'Design of a Michelson Interferometer for Quantitative Refraction Index Profile Measurements'
1998 / 60 pages / ISBN 90-407-1574-2
12. R.A.W.M. Henkes / J.L. van Ingen, 'Overview of Stability and Transition in External Aerodynamics'
1998 / IV + 48 pages / ISBN 90-407-1575-0
13. R.A.W.M. Henkes, 'Overview of Turbulence Models for External Aerodynamics'
1998 / IV + 40 pages / ISBN 90-407-1576-9

Series 02: Flight Mechanics

01. E. Obert, 'A Method for the Determination of the Effect of Propeller Slipstream on a Static Longitudinal Stability and Control of Multi-engined Aircraft'
    1997 / IV + 276 pages / ISBN 90-407-1577-7
02. C. Bill / F. van Dalen / A. Rothwell, 'Aircraft Design and Analysis System (ADAS)'
    1997 / X + 222 pages / ISBN 90-407-1578-5
03. E. Torenbeek, 'Optimum Cruise Performance of Subsonic Transport Aircraft'
    1998 / X + 66 pages / ISBN 90-407-1579-3

Series 03: Control and Simulation

01. J.C. Gibson, 'The Definition, Understanding and Design of Aircraft Handling Qualities'
    1997 / X + 162 pages / ISBN 90-407-1580-7
02. E.A. Lomonova, 'A System Look at Electromechanical Actuation for Primary Flight Control'
    1997 / XIV + 110 pages / ISBN 90-407-1581-5
03. C.A.A.M. van der Linden, 'DASMAT-Delft University Aircraft Simulation Model and Analysis Tool. A Matlab/Simulink Environment for Flight Dynamics and Control Analysis'
    1998 / XII + 220 pages / ISBN 90-407-1582-3

Series 05: Aerospace Structures and Computional Mechanics

01. A.J. van Eekelen, 'Review and Selection of Methods for Structural Reliability Analysis'
    1997 / XIV + 50 pages / ISBN 90-407-1583-1
02. M.E. Heerschap, 'User's Manual for the Computer Program Cufus. Quick Design Procedure for a CUt-out in a FUSelage version 1.0'
    1997 / VIII + 144 pages / ISBN 90-407-1584-X
03. C. Wohlever, 'A Preliminary Evaluation of the B2000 Nonlinear Shell Element Q8N.SM'
    1998 / IV + 44 pages / ISBN 90-407-1585-8
04. L. Gunawan, 'Imperfections Measurements of a Perfect Shell with Specially Designed Equipment (UNIVIMP)
    1998 / VIII + 52 pages / ISBN 90-407-1586-6

Series 07: Aerospace Materials

01. A. Vašek / J. Schijve, 'Residual Strenght of Cracked 7075 T6 Al-alloy Sheets under High Loading Rates'
1997 / VI + 70 pages / ISBN 90-407-1587-4
02. I. Kunes, 'FEM Modelling of Elastoplastic Stress and Strain Field in Centre-cracked Plate'
1997 / IV + 32 pages / ISBN 90-407-1588-2
03. K. Verolme, 'The Initial Buckling Behavior of Flat and Curved Fiber Metal Laminate Panels'
1998 / VIII + 60 pages / ISBN 90-407-1589-0
04. P.W.C. Provó Kluit, 'A New Method of Impregnating PEI Sheets for the *In-Situ* Foaming of Sandwiches'
1998 / IV + 28 pages / ISBN 90-407-1590-4
05. A. Vlot / T. Soerjanto / I. Yeri / J.A. Schelling, 'Residual Thermal Stresses around Bonded Fibre Metal Laminate Repair Patches on an Aircraft Fuselage'
1998 / IV + 24 pages / ISBN 90-407-1591-2
06. A. Vlot, 'High Strain Rate Tests on Fibre Metal Laminates'
1998 / IV + 44 pages / ISBN 90-407-1592-0
07. S. Fawaz, 'Application of the Virtual Crack Closure Technique to Calculate Stress Intensity Factors for Through Cracks with an Oblique Elliptical Crack Front'
1998 / VIII + 56 pages / ISBN 90-407-1593-9
08. J. Schijve, 'Fatigue Specimens for Sheet and Plate Material'
1998 / VI + 18 pages / ISBN 90-407-1594-7

Series 08: Astrodynamics and Satellite Systems

01. E. Mooij, 'The Motion of a Vehicle in a Planetary Atmosphere'
1997 / XVI + 156 pages / ISBN 90-407-1595-5
02. G.A. Bartels, 'GPS-Antenna Phase Center Measurements Performed in an Anechoic Chamber'
1997 / X + 70 pages / ISBN 90-407-1596-3
03. E. Mooij, 'Linear Quadratic Regulator Design for an Unpowered, Winged Re-entry Vehicle'
1998 / X + 154 pages / ISBN 90-407-1597-1

3021912

Cut-outs in pressurized aircraft fuselages are very sensitive to fatigue. This explaines the need for a design tool to enable the designer to perform a comprehensive design of this "difficult" structure. A menu-driven, highly interactive system for the design of cut-outs in pressurized fuselages is described. The design system is set up in such a way that maximum use is made of commercially available software. There are six basic options offered by the system:1) Easy and fast initial model generation; 2) Fully interactive, user friendly model editing; 3) Preparation for both geometrically linear and non-linear finite element calculations; 4) Preparation for a sensitivity analysis and graphical display of the results of such an analysis; 5) Carrying out "what-if" studies; 6) Pre- and post-processing of a design optimization.
This design system is set up as a special purpose design tool. This enables a substantial improvement in the design process, because the time needed for the modelling of the complex structure is greatly reduced. The design system is based on the pre- and post processor PATRAN. The necessary software is written almost entirely in PATRAN Command Language (PCL). Maximum use is made of the graphical capability of this software package. The finite element calculations, including sensitivity analysis and optimization, are performed by MSC/NASTRAN.

ISBN 90-407-1584-X
9 799040 715845

DELFT AEROSPACE
ENGINEERING & TECHNOLOGY