1989

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

ROCOPT - A USER FRIENDLY INTERACTIVE CODE TO OPTIMIZE ROCKET STRUCTURAL COMPONENTS

PREPARED BY:	WILLIAM K. RULE
ACADEMIC RANK:	ASSISTANT PROFESSOR
UNIVERSITY AND DEPARTMENT:	THE UNIVERSITY OF ALABAMA IN TUSCALOOSA DEPT. OF ENGINEERING MECH.
NASA/MSFC:	
LABORATORY:	STRUCTURES AND DYNAMICS
DIVISION:	STRUCTURES
BRANCH:	STRUCTURAL DEVELOPMENT
MSFC COLLEAGUE:	PEDRO I. RODRIGUEZ
DATE:	AUGUST 11, 1989
CONTRACT NO.:	THE UNIVERSITY OF ALABAMA IN HUNTSVILLE NGT-01-008-021

ABSTRACT

ROCOPT is a user-friendly, graphically-interfaced, microcomputer-based computer program (IBM compatible) that optimizes rocket components by minimizing the structural weight. The rocket components considered are ring stiffened truncated cones and cylinders. The applied loading is static, and can consist of any combination of internal or external pressure, axial force, bending moment and torque. Stress margins are calculated by means of simple closed form strength of material type equations. Stability margins are determined by approximate, orthotropic-shell, closed-form equations. A modified form of Powell's method, in conjunction with a modified form of the external penalty method, is used to determine the minimum weight of the structure subject to stress and stability margin constraints, as well as user input constraints on the structural dimensions. The graphical interface guides the user through the required data prompts, explains program options and graphically displays results for easy interpretation.

ACKNOWLEDGEMENTS

The author would like to thank Pete Rodriguez for initially suggesting this project and helping to carry it out. The author would also like to thank Clyde Nevins for his support during the course of this effort, and Stan Jones for the encouragement to apply for this fellowship. Thanks are also due to Jerry Karr, Frank Six, and Mike Freeman for running a very enjoyable and productive summer faculty program. Finally, the author would like to thank his wife Audrey and daughter Amethyst for enduring the separation from our home and friends in Tuscaloosa this summer.

TABLE OF CONTENTS PAGE

Abstract XXV-i

Acknowledgements XXV-ii

Introduction XXV-1

Objectives XXV-4

Conclusions and Recommendations XXV-5

1. Description of ROCOPT Programs
   - 1.1 Getting ROCOPT Running XXV-7
   - 1.2 ROCOPT Batch Program XXV-7
   - 1.3 HEADER Program XXV-8
   - 1.4 MATINPUT (Material Type Input) Program XXV-8
   - 1.5 STRINPUT (Structure Type Input) Program XXV-8
   - 1.6 LODINPUT (Load Input) Program XXV-9
   - 1.7 CONE and CYLINDER Optimization Programs XXV-9
   - 1.8 OUTPUT Program XXV-10
   - 1.9 HELP and MANUAL Programs XXV-10

2. Stress Margin Calculation
   - 2.1 Cylinder Stress Margin XXV-12
   - 2.2 Cone Stress Margin XXV-14

3. Stability Margin Calculation
   - 3.1 Cylinder Stability Margin XXV-15
   - 3.2 Cone Stability Margin XXV-16

4.Optimization Technique XXV-17

References XXV-23

Appendix - Program Listings XXV-24

INTRODUCTION

The goal of this project was to develop a user-friendly, graphically-interfaced, microcomputer-based computer program (IBM compatible) to optimize rocket structural components by minimizing the total structural weight. The resulting program is called ROCOPT. It consists of a number of separate executable programs written in Microsoft QuickBASIC that are linked together by means of a DOS batch file. The program is designed to allow the user to rapidly determine a reasonable preliminary design for ring stiffened conical and cylindrical rocket structural components subjected to any static combination of internal or external pressure, axial force, bending moment, and torque. User input constraints on structural dimensions are also applied. Stress margins are calculated by means of simple closed form strength of materials type equations. Stability margins are determined by approximate, orthotropic-shell, closed-form equations. Thus, the preliminary design will require checking by means of a detailed finite element model.

QuickBASIC was chosen because of it's excellent programing environment and built-in graphics capabilities. QuickBASIC can run in an interpretive mode which greatly facilitates debugging, and it can also run in an compile mode to generate executables that do not have to be run from within the BASIC environment. Many members of the Structural Design Branch are already familiar with BASIC. Thus, in-house maintenance and enhancement of ROCOPT should be feasible.

ROCOPT consists of a number of stand-alone executable programs (each of which has a separate function) rather than a single large program. This was done for three main reasons:

1. Each program has a separate function. This made the programing task easier by breaking a large program into a number of smaller ones that could be developed and debugged separately. This technique should also simplify program maintenance and enhancement.

2. Any of the ROCOPT programs can be replaced by another program written in a different programming language. Thus, personnel in charge of enhancement are not necessarily required to know BASIC.

3. ROCOPT can be made arbitrarily large since only one of

the ROCOPT programs is loaded into the memory at any time. A single program doing all the functions of ROCOPT would have to all be loaded in the computer memory at the same time. This could cause memory shortage problems if ROCOPT should grow.

The programs of ROCOPT are conveniently linked together by a DOS batch file. The batch file controls the sequence of program execution, allows for rerunning all the programs without restarting, and permits a graceful halting of the program sequence should an error occur. The batch file also allows for the user to quit the program at any time.

A modified form of Powell's method (a zero order method) is used for the optimization algorithm. Instead of using the identity matrix as the initial matrix of search vectors as Powell suggested, a matrix of random vectors (each normalized such that the maximum magnitude of any vector component is unity) is used instead. This modification appears to improve performance, on the average, for any given run. It has the added advantage that if multiple runs are made then entirely new search paths are followed. This is useful for verifying that the minimum structural weight obtained is indeed the global minimum. This is so if several runs end up selecting the same point in the design space as the optimum.

A modified form of the exterior penalty method is used to constrain the solution to having nonpositive stress and stability margins of safety. The method is modified as follows. The traditional penalty factor is replaced by a penalty factor multiplied by the weight of the structure. The constraint functions (stress and stability margins) are always of order unity and thus some scheme is required to scale the constraints up to the magnitude of the structural weight. This scheme avoids the problem of how to reliably select a suitable scaling factor for the penalty part of the pseudo-objective function. Also, the constraint functions are typically squared. This is not done here.

A standard strength of materials approach is used to calculate the stress margin. Here the stress margin is defined as the maximum von Mises (also called "effective") stress divided by the allowable stress, minus unity. Thus, a component with a stress margin greater than zero is over stressed. Stress margins are calculated for both the shell and the ribs. Similarly, a stability margin is calculated based on allowable stresses (both circumferential and longitudinal) calculated from approximate orthotropic thin shell buckling equations. A relatively thin shell with relatively small closely spaced ribs is the assumed structural type for all margin calculations. The exact definition of what is meant by small and the effects of deviations from these requirements are difficult to assess without going to a detailed finite element model of the proposed structure.

A "strip-chart" of change in structural weight versus iteration number is shown on the screen as the optimization proceeds to allow the user to monitor the process. The optimization process can be stopped at any time by simply pressing the F1 function key. A cross section of the optimized structure is drawn approximately to scale on the screen for the user to visually assess the suitability of the optimum structure. Also, all the user inputs and the calculated results are echoed to a file for later inspection.

OBJECTIVE

The objective of this project was to develop a user-friendly, graphically-interfaced, microcomputer program (IBM compatible) to determine the minimum feasible structural weight of statically loaded, ring-stiffened, conical and cylindrical rocket components using approximate closed form stress and stability equations.

CONCLUSIONS AND RECOMMENDATIONS

The following conclusions were reached during the course of this project:

1. Microsoft QuickBASIC is an ideal language for PC-based interactive engineering analysis codes.

2. Using a DOS batch file is a convenient way to link together a series of executable programs. Large programs can be developed because only one of possibly many executable programs is loaded into the memory at a time. The executable programs can be written in any computer language capable of producing an executable code. This allows code developers to use any computer language they wish for their part of the total package.

3. There is a large amount of uncertainty with respect to predicting buckling of ring stiffened cones and cylinders subjected multiple loads using closed form analytical equations.

4. The optimization scheme developed during the course of this project appears to be relatively easy to program, robust, and efficient.

5. ROCOPT appears to be capable of successfully optimizing ring stiffened cone and cylinder structures within a reasonable amount of run time. A typical ROCOPT run takes less than five minutes. However, multiple runs should be made to ensure that the global optimum (lowest feasible weight) has been located.

The recommendations are as follows:

1. Further development work on the stability equations should be performed.

2. Additional rib cross section type options should be added to ROCOPT. Currently, rectangular and I-beam cross sections are treated.

3. The capability to handle stringers should be added to ROCOPT.

4. The capability of treating composite materials should be incorporated into ROCOPT.

5. An option to automatically create an MSC/PAL input deck of the optimized structure should be provided. This would facilitate the checking of the optimized structure by means of a detailed finite element model.

6. Detailed finite element models of ROCOPT optimized structures should be built and analyzed to confirm that stress and stability constraints are not violated.

1. Description of ROCOPT Programs

1.1 Getting ROCOPT Running

The following procedure should be followed to run ROCOPT. First, create a subdirectory on your hard disk. Then, copy all the files contained on the ROCOPT floppy disk(s), which are in the envelope at the back of the manual, into your subdirectory. Get into the subdirectory and start the program by typing ROCOPT. The program will automatically prompt you for inputs by making selections from menus. Menu selections are made by using the up and down arrows until the desired choice is highlighted and then pressing RETURN. A brief description of each highlighted menu choice is given across the bottom of the screen. The purpose of the current program module of ROCOPT is displayed across the top of the screen. Your computer should have a math coprocessor and EGA or VGA color graphics capability.

1.2 ROCOPT Batch Program

ROCOPT consists of a DOS batch program (ROCOPT.BAT) that runs a series of other programs written in Microsoft QuickBASIC. These programs and their associated data files are illustrated in Figure 1. An overview of these programs will now be given. Listings of all ROCOPT programs are given in the Appendix.

The batch program ROCOPT.BAT sequentially runs the other programs of ROCOPT. The batch program checks for the existence of a program before attempting to run it. If the program is not present on the directory, an error message is printed and execution is stopped. The inputs and outputs of the previous run are stored (by program OUTPUT) in a file called OLDINPUT.DAT. The batch program checks if this file exists. If not, it creates one by copying the file NOOLDINP.DAT which contains the statement "NO PREVIOUS RUN DATA AVAILABLE!".

The user always has the option to quit running ROCOPT while running any of the ROCOPT programs. If the user elects to do so, a file called QUIT.DAT is created. The batch program checks for the presence of this file after every program is run. If detected, the batch program will stop itself. If an error is detected while running a program then a file named ERROR.DAT is created on the directory. If this file is detected, then the batch program will output an error message and then the batch program will stop itself.

If the user elects to optimize a cone or a cylinder then the empty files CONE.DAT and CYLINDER.DAT are created (by program STRINPUT), respectively. The batch program will execute the appropriate program (CONE or CYLINDER) depending on which of these files is present.

If the user decides to rerun the analysis without exiting the batch program then the empty file RERUN.DAT is created (by program OUTPUT). When the batch file detects the presence of RERUN.DAT, it loops back to the start of itself to begin again (without rerunning program HEADER).

1.3 HEADER Program

The first program run by the batch program is called HEADER. This program simply displays the name and version of the program as well as a brief description of the purpose of the program. No calculations are performed, and no data is input here.

1.4 MATINPUT (Material Type Input) Program

The next program executed is MATINPUT. This program inputs the following characteristics of both the shell and rib materials: material name (15 characters maximum), the elastic modulus, the shear modulus, the allowable stress, and the density. Any consistent set of units can be used. The user has the option of using values from a previous run, selecting values from a database file (MATERIAL.DAT), or entering data directly through the keyboard. The user has the option of writing the directly entered data into MATERIAL.DAT. MATERIAL.DAT is a simple ASCII file that can be modified by most word processing programs. MATINPUT also prompts the user for a title for the current run and samples the computer's clock for date and time. All input data is written to file INPUT.DAT for later use.

1.5 STRINPUT (Structure Type Input) Program

STRINPUT is the next program executed. This program prompts the user for the type of shell (cone or cylinder) and the type of rib (rectangular cross section or I-beam). Minimum and maximum dimensions are also prompted for. Zero minimum dimensions are not allowed. If the input maximum dimension is less than the minimum dimension then the program assumes that there is no restriction on the maximum magnitude of that dimension. The user has the option of selecting previous shell types, rib types, and dimensions, which are

ORIGINAL PAGE IS
OF POOR QUALITY

contained in file OLDINPUT.DAT. A graphical interface is provided to help the user with the input. All input data is written to file INPUT.DAT for later use.

1.6 LODINPUT (Load Input) Program

LODINPUT is run next to input information on the static loads. The loads include: uniform pressure, axial force, bending moment, and a torque. The uniform pressure is assumed to be applied in such a manner as to produce both circumferential and longitudinal stress. The user has the option of using the loads of the previous run, which are contained in file OLDINPUT.DAT. All input data is written to file INPUT.DAT for later use.

1.7 CONE and CYLINDER Optimization Programs

Next ROCOPT will run program CONE if a cone is to be optimized, otherwise program CYLINDER is run. Both programs are very similar in nature. They read the user supplied data in file INPUT.DAT and then prompt the user for parameters relating to the optimization process. Then CONE or CYLINDER perform stress margin, stability margin, and optimization calculations. Detailed descriptions of how these calculations are performed are given in sections 2, 3, and 4 of this report, respectively. The goal of these programs is the find the structural configuration with the lowest possible weight that is stable (no buckling) and not over stressed. User input constraints on minimum and maximum structural dimensions must also be satisfied. The change in the total structural weight as a function of iteration number is graphically shown on the screen, in a continuously updated bar chart format, to show the user how the optimization is proceeding. Also, the total structural weight, the stress margin, and the stability margin are shown on the screen above the bar chart periodically to further indicate the health of the optimization process.

The user can safely halt the iteration process at any time by simply pressing the F1 function key. This may be desirable if the bar chart of change in weight versus iteration number shows that there has been no change in the structural weight after many iterations. A typical optimization run takes less than 5 minutes to execute on a 80286 class of machine. Details of the final optimized structure are written to file INPUT.DAT for later output.

As discussed in section 4, CONE and CYLINDER use random numbers in their optimization algorithm. The random number generator is programmed to use the number of seconds from midnight on the computer's clock (suitably scaled) as a seed. Each time CONE or CYLINDER is run, entirely different numbers will be used in the optimization algorithm. Thus, it is desirable to make several runs of ROCOPT with the same structural configuration to ensure that the lightest feasible structure has been found. It is not uncommon for any optimization process to get trapped in a local minimum. Making several runs from different points in the design space is a way to defeat this problem.

1.8 OUTPUT Program

The last program the batch program runs is called OUTPUT. This program performs the following functions:

1. Reads the user input data and calculated results contained in data file INPUT.DAT and writes them in ASCII format to a new file that program OUTPUT creates that is named by the user. The file can be in a different directory from that of the ROCOPT PROGRAM by specifying a path before the file name. For instance, specifying the file name A:\RESULTS\SHUTTLEC.OUT will cause the results to be written to file SHUTTLEC.OUT located in directory A:\RESULTS. If the file already exists, then it will be overwritten. OUTPUT also offers the user the option of printing the file out. The file is formatted reasonably nicely and could perhaps be included directly in some reports. Alternatively, most word processors could read the file into a report document, where the layout of the file could be refined as required.

2. Displays the optimum structural dimensions, margins, and weight on the screen. Also, OUTPUT displays a cross section of the optimized structure on the screen approximately to scale. The purpose of these options is to allow the user to quickly assess if the optimum structure appears to be reasonable when compared with past experience.

3. OUTPUT also allows the user to restart the analysis. The built in capability of all the ROCOPT programs to use the user inputs of the previous run, allows for rapid reruns when there are no modifications or slight modifications to the input data.

1.9 HELP and MANUAL Programs

Finally, there are two additional auxiliary programs in the ROCOPT family of programs that work outside of ROCOPT and are purely for helping the user run ROCOPT more successfully. One is called HELP. This program simply allows the user to select topics of interest from a menu and then to access the ASCII text file containing that information. HELP allows the user to page through the text file by means of the PAGE UP and PAGE DOWN keys as well as go to the start and end of the text file by pressing the HOME and END keys. The text files are currently empty - the user must use a text editor to add information to these files as pertinent information becomes available. Thus, HELP simply provides an organized way of storing data related to ROCOPT. For instance, the HELP menu selection SHELL could be tied to a text file containing information such as the shell thicknesses and shell materials used previously for the Saturn V, Titan, Shuttle SRM, and so forth. The topics displayed by the HELP menu and their associated text file names are stored in a file called HELP.MNU. This ASCII file can be easily modified by the ROCOPT user to add more topics or remove some of the topics currently in place. The format of this file consists of groups of three records relating to each topic. A typical entry would be as follows:

```
                                  .
                                  .
                                  .
SHELL
This file contains previously used shell thicknesses.
c:\HELPFILES\SHELL.HLP
                                  .
                                  .
                                  .
```

For this case, SHELL (15 characters max.) will be the menu prompt, the second record (78 characters max.) will appear at the bottom of the screen when SHELL is the highlighted menu selection, and the associated text file (could be any legal DOS name) is named SHELL.HLP and is located in subdirectory (could be any legal subdirectory) c:\HELPFILES.

This help utility can be accessed by simply typing HELP.

The second auxiliary program is called MANUAL. This program allows the user to page through an unformatted ASCII version of this document called (ROCOPT.DOC) interactively on the computer screen. This will conveniently avoid the problem of locating a hard copy of the manual if a question about ROCOPT should arise. Also, the user is free to customize ROCOPT.DOC with additional information. Any word processor or editor capable of working with ASCII files could be used for this purpose. This auxiliary program may be accessed by typing MANUAL.

2. Stress Margin Calculation

2.1 Cylinder Stress Margin

The stress margin is defined here as the largest von Mises stress (also called effective stress) divided by the allowable stress for the material, minus one. Thus, a positive stress margin implies that the component is over stressed. The von Mises failure theory is only applicable to isotropic metallic materials that are behaving in a ductile manner [1]. Typically the allowable stress is determined by dividing the yield stress by an appropriate factor of safety. Program CYLINDER calculates the stress margin for the shell material and the rib material and then uses the largest of these two in the optimization calculations.

The circumferential (hoop) stress in the shell wall, $\sigma_{y,shell}$, was determined as follows. The traditional (no rings) stress formula is [1]:

$$\sigma_{y,shell} = \frac{PD}{2t} \qquad (1)$$

where P is the pressure (positive if internal), D is the mean diameter, and t is the shell thickness. Tensile stresses are positive. Here, this equation must be modified to account for the ribs. It is assumed that the ribs will sustain the same strain level as that of the shell throughout their entire depth (no flexure). For this assumption to be accurate the diameter of the cylinder must be much larger than the depth of rib and the ribs must be fairly closely spaced (so shear lag will not occur). Also, since the ribs could be fabricated from a different material, the rib cross sectional area, A_R, must be multiplied by the modular ratio, M, to convert the rib area to an effective area of rib (as if it was made from the shell material). The modular ratio is E_R/E_S, where E_R and E_S are the elastic moduli of the rib and the shell, respectively. Let the number of ribs per unit length of shell be N. Modifying (1) to account for the ribs gives:

$$\sigma_{y,shell} = \frac{PD}{2t + 2A_R MN} \qquad (2)$$

It is assumed that P is the only applied load causing stress in the ribs. The rib stress in the circumferential direction may be calculated by multiplying the strain in the shell, which is assumed to be equal to the strain in the rib, by the elastic modulus of the rib:

$$\sigma_{y,rib} = E_R(\sigma_{y,shell}/ E_S) \qquad (3)$$

Stress in the shell in the longitudinal direction is generated by the applied pressure P, the axial force F, and the bending moment B. Here the sign convention is such that a positive P will generate tensile stress, as will a positive F. B will generate both tensile and compressive stress, regardless of sign, and thus both cases must be considered in the calculations. P, F, and B are assumed to generate no stress in the ribs in the longitudinal direction. The stress due to P is simply the pressure times the total longitudinal cross sectional area of the cylinder divided by the longitudinal cross sectional area of cylinder wall material. The axial force creates a stress equal to F divided by the longitudinal cross sectional area of shell material. The bending stress can be calculated from the flexure formula. Thus the shell stress in the longitudinal direction, $\sigma_{x,shell}$, is given by:

$$\sigma_{x,shell} = \frac{PD}{4t} + \frac{F}{\pi Dt} \pm \frac{4B}{\pi D^2 t} \qquad (4)$$

The longitudinal rib stress is assumed to be equal to zero.

The applied torque T, is assumed to generate a shear stress in the shell material (uniform through the thickness), $\tau_{xy,shell}$, but to have no effect on the ribs. The classical strength of materials formula was used to calculate the shear stress:

$$\tau_{xy,shell} = \frac{2T}{\pi D^2 t} \qquad (5)$$

The sign of the shear stress does not matter as it is squared in the von Mises stress formula.

After, the stress components have been calculated, the von Mises (effective) stress, $\sigma_{E,shell}$, is calculated for the shell material [1]:

$$\sigma_{E,shell} = (\sigma^2_{x,shell} + \sigma^2_{y,shell} - \sigma_{x,shell}\sigma_{y,shell} + 3\tau^2_{xy,shell})^{1/2} \qquad (6)$$

In general, $\sigma_{E,shell}$ will have to be calculated twice since B generates both tensile and compressive stress. The largest of these two is used to calculate the stress margin for the shell material. The von Mises stress for the rib is simply the absolute value of of $\sigma_{y,rib}$.

2.2 Cone Stress Margin

Precisely the same assumptions that are applied to the calculating the stress margins in the cylinder are applied to calculating the stress margins in the cone. Stress equations applicable to the cone can be obtained from the cylinder equations by replacing t with t/cos(cone angle). Here, the cone angle is the angle the side of the cone makes with the longitudinal axis (a cylinder has a cone angle of zero). Also, all stresses must be checked at both ends of the cone as there are now two diameters to consider.

ORIGINAL PAGE IS
OF POOR QUALITY

3. STABILITY MARGIN CALCULATION

3.1 Cylinder Stability Margin

The stability margin is defined as the applied compressive load divided by the calculated critical buckling load minus one. Thus. a stability margin greater than zero implies that buckling could occur.

The stress levels that could cause instability (buckling) of the cylinder are calculated from an approximate orthotropic cylinder theory exactly as presented in Ref.[2] except that a factor of safety of 2 against buckling was incorporated. This safety factor is to allow for the approximations inherent in the theory and also to account for small imperfections in the structural geometry. The theory assumes that the ribs are sufficiently closely spaced so that the shell can be modeled as an orthotropic continuum. The equations employed by ROCOPT can be obtained from [2].

There are many approximations associated with these calculations. Thus, the resulting final optimum structure should be checked with a detailed finite element model.

The stability calculations involve the positive integer parameters m and n, which are the number of longitudinal, and circumferential buckles (pop outs or pop ins), respectively, in the buckled shell wall. From physical considerations (no kinks in the shell wall) n must consist of even integers. Also, [2] recommends that n=2 not be used. The standard procedure is to vary these parameters until a minimum theoretical buckling load is determined.

The computer program does this in an iterative manner. During each optimization iteration, the program tries nine combinations of three adjacent, feasible m-values and n-values, and stores the m,n combination that produces the lowest calculated buckling load. During the next optimization iteration, the procedure is repeated with previously stored m and n values at the middle of the sets of m and n trial values. For instance, if the previously stored m and n values were 5 and 8, respectively, then during the next optimization iteration the following values would be tried: m=[4,5,6] and n=[6,8,10]. Three trial values were selected to give m and n the freedom to either increase or decrease with each iteration with a minimum number of m,n trial combinations.

The smallest dimension of a buckle is prevented from being smaller than twenty times the shell thickness. The computer program sets limits on the maximum values of m and n to apply this limit. The author developed this constraint to ensure that the calculated buckled configuration is physically reasonable. Experimental results of testing full scale shell structures do not indicate the formation of very fine buckles.

3.2 Cone Stability Margin

As suggested by [3], cone buckling loads are evaluated by first "converting" the cone into an "equivalent" cylinder and then using cylinder stability equations. The equivalent cylinder has a diameter equal to the mean diameter of the cone and a length equal to the side wall length (axial dimension/cos[cone angle]) of the cone. This approximation reflects the desire for simple design equations and the lack of a wide spectrum of experimental data to verify more sophisticated cone stability equations. However, this approach should be sufficient for preliminary design calculations, which is the intended purpose of ROCOPT.

4. Optimization Technique

Many optimization techniques have been reported in the literature. These iterative schemes vary from the simple to the very complex. The simple schemes are typically easy to program, are very simple and reliable in the sense that they nearly always work, and require little or no specialized help from the user. Unfortunately, simple schemes are usually the most computationally expensive. Complex optimization techniques are difficult to program, do not always work, and in many cases require careful monitoring by an experienced optimization expert. However, complex optimization schemes can be orders of magnitude less computationally expensive than the simple schemes, as well as produce more accurate results.

It is anticipated that the average ROCOPT user will not be acquainted with optimization theory. Also, most ROCOPT users will probably be willing to wait a few minutes longer for assured results, rather than run the risk of obtaining incorrect results or no results at all very quickly. These considerations led to the choice of a relatively simple optimization technique.

ROCOPT uses Powell's method combined with the exterior penalty method [4]. The author modified both these methods for this application to increase reliability and efficiency. The optimization technique used will now be discussed.

The purpose of ROCOPT is to find the structure with the lowest weight that does not violate any of the constraints. The weight of the structure is called the objective function. The purpose of any optimization technique is to minimize the objective function subject to the constraints.

There are two different types of constraints that must be treated by ROCOPT. The design variables (shell thickness, rib depth and so forth) must be made to stay within their user defined minimum and maximum ranges. This type of constraint is called a side constraint. Also, the stress and stability margins of the final design must be less than or equal to zero. These are termed inequality constraints.

Side constraints are most efficiently handled directly. If any design variable is found to be outside of it's allowable range, the design variable is simply increased or decreased as appropriate, until it is back within it's allowable range. This is done before additional optimization calculations are performed. Thus, throughout the optimization iterations, the design variables are explicitly forced to remain within their allowable ranges. This approach is feasible because each side constraint is a function of a single design variable.

Inequality constraints are usually functions of many design variables and thus there is no direct way to treat a constraint violation. One approach is to add violated (greater than zero) inequality constraints to the objective function in some fashion. The objective function so modified is typically called the pseudo-objective function (POF). This converts the problem from one of minimizing an objective function subject to inequality constraints, to one of minimizing an unconstrained pseudo-objective function. The form of the pseudo-objective function developed by the author is:

$$\mathrm{POF} = (\mathrm{weight})[1 + r_p(\frac{\mathrm{stress}}{\mathrm{margin}} + \frac{\mathrm{stability}}{\mathrm{margin}})] \qquad (7)$$

Note, the margins are only allowed to enter the above equation if they are greater than zero (constraint violated). r_p is a positive parameter called the penalty parameter. Notice that a large value for the penalty parameter will provide a big incentive to the optimizer to adjust the design variables to alleviate the constraint violations, at the expense of increasing the structural weight. However, since the margins are actually multiplied by the weight times the penalty factor, the constraint violations tend to be removed in a manner that increases the weight the least.

The magnitude of the penalty parameter is a user controlled input. A value of 10 is suggested. A relatively small penalty factor is used in the initial phase of the optimization process so that the optimizer will effectively be working on reducing both the structural weight and the constraint violations. If the penalty parameter is too large initially, then too much emphasis will be placed on eliminating the constraints at the expense of greatly increasing the structural weight. This will result in an inefficient optimization process because many iterations will be required to find the lowest possible structural weight, after the constraint violations (and thus the penalty factor effect) have been removed. The recommended procedure is to start with a relatively small penalty factor

ORIGINAL PAGE IS OF POOR QUALITY

and then increase it as the optimization process continues [4]. ROCOPT increases the user input penalty factor by a factor of 100 by the end of the iterations. A high penalty factor is used at the end to ensure that the final design will be feasible (no constraints violated). The user is free to try various initial values for the penalty parameter to try improving the optimization efficiency.

Having set up the POF, the problem now becomes one of finding the unconstrained minimum of this function.

The design variables associated with this optimization problem vary greatly in magnitude. For instance, the shell thickness could be 0.1 inches and the shell length could be 1000 inches, a difference of four orders of magnitude. Because of this, using these design variables directly is inconvenient and can lead to numerical problems. Thus, it is advisable to work with nondimensionalized design variables that have been scaled to a magnitude of unity. This is similar to the rational for using isoparametric elements in finite element analysis.

To achieve this nondimensionality, ROCOPT works with percentage changes in design variables. This conveniently provides a uniform treatment for all the design variables, regardless of magnitude. It also provides a simple way of controlling the amount of change in the design variables from one optimization iteration to the next. If the maximum allowable percentage change is too large, the optimizer could thrash back and forth around the optimum design point without ever converging to it. Alternatively, if the maximum allowable percentage change is too small then it could take an infeasible number of iterations to get to the minimum, or the optimizer could get "stuck" in a local minimum of the pseudo-objective function before getting to the global minimum.

ROCOPT calls the maximum allowable change in the nondimensionalized design variable magnitudes the "search magnitude parameter". This is a user controlled input parameter. A value of 0.4 (equivalent to a 40% change) is recommended. ROCOPT is designed to reduce the magnitude of the search magnitude parameter as the optimization process proceeds. The final value will be 1/100 of the initial value. The idea here is to allow large changes in the design variables initially, to quickly get into the vicinity of the global minimum in the design space, and then use finer steps to precisely locate the global minimum. The user is free to change this parameter to try to improve optimization efficiency.

The initial values for the design variables are randomly selected between their minimum and maximum values. If no maximum value is provided for a design variable, then a random value between the minimum value and four times that value is used. The number of seconds from midnight on the computer's clock is used as the seed for the random number. Thus, a series of ROCOPT runs will all start from different points in the design space. A global minimum is indicated if essentially same the optimum solution is obtained from a series of ROCOPT runs.

Finally, there is the problem of determining the search vectors. A search vector is a vector of changes to be made in the design variables to seek minimization. A common approach is to use the negative of the gradient vector. However, this method is somewhat computationally expensive and may not be reliable in cases where there are many constraints (discontinuous gradients) or where the gradient function changes rapidly throughout the design space. The literature contains many variations of the gradient search method as well as a vast array of simpler and more complex methods.

The method chosen here for search vector selection is based on Powell's method [4]. This is a first order method that does not require the calculation of the gradient vector. The author modified this method in an attempt to increase efficiency. Initially, a number of search vectors equal to the number of design variables are created. The components of these vectors are random numbers between -1 and +1. The components of each random search vector are then scaled, such that the largest component has a magnitude of unity. These vectors are stored as columns of a "search matrix". Next, the POF is evaluated at the current point in the design space and at design points given by +/- the search magnitude parameter times the first column of the search matrix. If either of the + or - design points has a POF less than the POF of the current design point, then the design point corresponding to the lowest POF will become the new design point. Otherwise, the design point does not change. The search vector multiplier (+/- search magnitude parameter or zero) used with the search vector is stored for later use. This procedure is then repeated with the remaining columns of the search matrix.

A new search vector is created after using all of the search vectors in the search matrix. This new vector is created by vectorially adding together all of the search vectors times their search vector multipliers. The new search vector is a vector sum of previous successful search vectors since unsuccessful search vectors have search vector multipliers of zero. Thus, the new search vector represents (stores) the trend of the optimization process. The new search vector is scaled such that the magnitude of it's largest component is unity and then is used to replace the first column of the search matrix. The procedure is repeated, a new search vector is determined, and then used to replace the second column of the search matrix, and so forth until only the last column of the original search matrix remains untouched. Then an entirely new search matrix is created using the random number generator, and the process continues.

If at any time in the iterative process, a new search vector has a magnitude of zero (implying all current search directions are not beneficial), then a new random search matrix is created immediately. The random number generator uses a seed based on the number of seconds from midnight on the computer's clock. Each successive run of ROCOPT will use a different set of search vectors. Thus, it is advisable to make several successive runs of ROCOPT with identical inputs to ensure that the lowest possible structural weight has been found.

The number of search matrices generated is governed by a user input parameter called the "iteration parameter". The number of random search matrices generated is equal to the number of design variables times the iteration parameter. The suggested value for the iteration parameter is 20. However, the user has the option of shutting off the optimization process at any time by pressing the F1 function key. Changes in the structural weight and the margins are displayed on the screen (as the iterations progress) to help the user decide on an appropriate time to shut off the iterations.

Each time a new random search matrix is created. the penalty parameter is multiplied by a factor such that, by the time the optimization process is completed, the penalty factor will be 100 times its initial value. The same is true for the search magnitude parameter except that it's magnitude is reduced by a factor of 100. In both cases, this assumes that the user did not stop the iterations by pressing the F1 key.

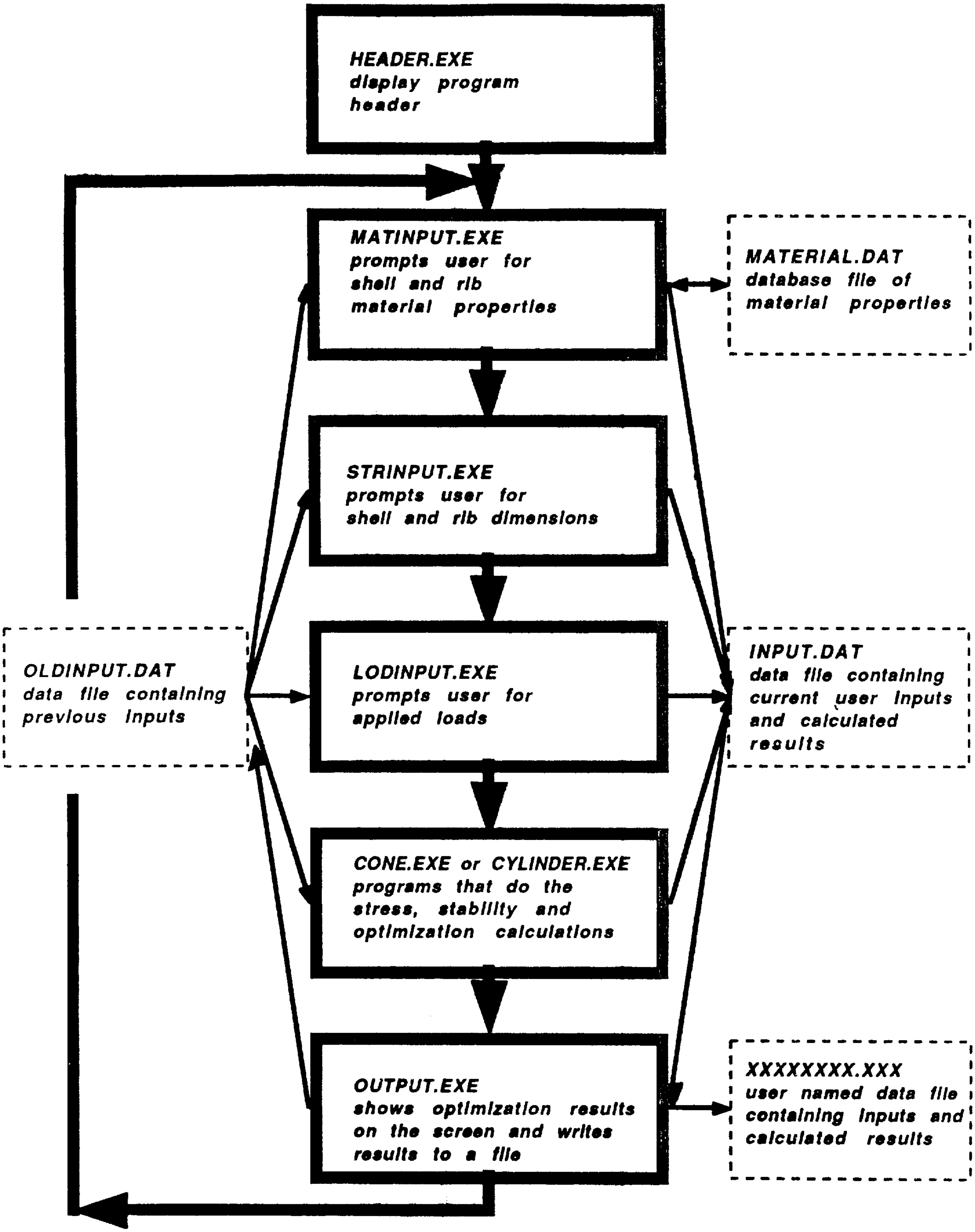
HEADER.EXE
display program header
MATINPUT.EXE
prompts user for shell and rib material properties
MATERIAL.DAT
database file of material properties
STRINPUT.EXE
prompts user for shell and rib dimensions
OLDINPUT.DAT
data file containing previous inputs
LODINPUT.EXE
prompts user for applied loads
INPUT.DAT
data file containing current user inputs and calculated results
CONE.EXE or CYLINDER.EXE
programs that do the stress, stability and optimization calculations
OUTPUT.EXE
shows optimization results on the screen and writes results to a file
XXXXXXXX.XXX
user named data file containing inputs and calculated results

Figure 1 ROCOPT Data Flow Diagram

ORIGINAL PAGE IS
OF POOR QUALITY

REFERENCES

1. Cook, R.D., and Young, W.C.,"Advanced Strength of Materials," Macmillan Publishing Company, New York, 1985.

2. Weingarten, V.I., Seide, P. and Peterson, J.P., "Buckling of Thin-Walled Circular Cylinders," NASA SP-8007, 1968.

3. Weingarten, V.I. and Seide, P., "Buckling of Thin-Walled Truncated Cones," NASA SP-8019, 1968.

4. Vanderplaats, G.N., "Numerical Optimization Techniques for Engineering Design: With Applications," McGraw-Hill Book Company, New York, 1984.

APPENDIX - PROGRAM LISTINGS