Design Improvements using Design Sensitivity Analysis Capability of MSC/NASTRAN

by

Dr. R. Narayanaswami

President, Computerized Structural Analysis & Research (CSAR) Corporation

Northridge, California

INTRODUCTION

The capability to obtain design improvements in an automated fashion is useful, and often necessary, to design engineers. Engineers have, during the past thirty years, developed methodologies for automated design that vary in complexity from the simple stress-ratio algorithms of 'Fully Stressed Design' (FSD) to the more complex algorithms based on mathematical programming techniques. Such methods are collectively known by various names as "minimum weight design", "structural optimization", "structural synthesis", etc. The evaluation of the sensitivity matrices (gradient of the constraints with respect to the design variables) is an important step in these methods. MSC/NASTRAN has introduced the design sensitivity analysis capability in Version 63 to evaluate the sensitivity matrices. This paper discusses the development of two software products CSAR/SIZING and CSAR/ØPTIM that provide the MSC/NASTRAN users with the capability to obtain design improvements in an automated fashion.

THEORETICAL DISCUSSION

The theoretical details of the various methods commonly used for minimum weight design of structures are now generally well understood and a large volume of literature (for example, references 1-10) is available that explain those methods with clarity. No attempt will be made herein to derive those equations again; rather, the interested reader may consult the references given at the end of the paper. A brief survey of the techniques, however, is presented below.

The intuitive feeling that all structural members of a structural model, if fully stressed under at least one loading condition, will have optimum sizes and hence the structural model will be at its 'minimum weight' is the basis for "fully stressed design". Of course, there is no mathematical proof that this indeed is the case (and, in fact, it is now known that there is no guarantee that fully stressed design will always yield a minimum weight design). Nevertheless, this method has found acceptance among design engineers due to its simplicity and ease of use. Moreover, in most cases, the design engineer is generally interested in obtaining a much improved design (where members are more or less stressed to its allowable limit) than the one he started with and not necessarily in iterating to the absolute possible minimum weight. The algorithm used is, for the case of an axially loaded rod element,

$$A_{i+1} = \frac{\sigma_i}{\sigma_a} A_i \qquad (1)$$

where A_{i+1} = The area at iteration i+1

A_i = The area at iteration i

σ_i = The stres at iteration i

and σ_a = The allowable stress

Similar equations can be written for other structural elements like plate elements or shear panels.

Equation (1) and similar other equations are known as the resizing algorithms of FSD.

Observing the fact that thermal stresses are insensitive to structural sizes (expression for thermal stress in a rod is $E\alpha t$, where E is elastic modulus, α is thermal expansion coefficient and t is the temperature change), Adelman and Narayanaswami (Reference 11) have developed the thermal fully stressed design (TFSD) technique. The TFSD algorithm is, for uniaxially stressed members,

$$A_{i+1} = \frac{\sigma_{Mi}}{\sigma_{a,M} - \sigma_{Ti}} A_i \tag{2}$$

where σ_{Mi} = Stress due to mechanical loads acting alone

σ_{Ti} = Stress due to thermal loads acting alone

$\sigma_{a,M}$ = Tensile or compressive allowable stress, depending on the sign of σ_M

For problems having substantial thermal stress, TFSD was found, in Reference 11, to converge in far fewer iterations than FSD.

The resizing algorithms of FSD and TFSD help to perform strength sizing of structural members. Where design improvements are needed with constraints on displacement, vibration frequency or flutter, other methodology is needed. Two schools of thought have evolved in the development of such methodologies, (i) the mathematical programming methods and (ii) the optimality criteria methods.

Mathematical Programming Based Techniques:

In designs based on mathematical programming techniques, an objective function (say, the weight of the structure) is minimized subject to a given set of constraints (limits on stress, displacement or minimum and maximum sizes of members).

The problem of weight minimization of a structural model will generally be stated as follows:

Minimize
$$W(\vec{V}) = \sum_{i=1}^{N} w_i V_i \tag{3}$$

where w_i = the weight of member i for unit value of V_i, the design variable (for example, specific weight times length of a truss member) subject to behavioral constraints (behavioral constraints impose limitations on stresses, displacements, natural frequencies, etc.)

$$g_q(\vec{V}) \geq 0 \qquad q = 1,2,\ldots,Q \tag{4}$$

and/or side constraints

$$V_i^{(L)} \leq V_i \leq V_i^{(U)} \qquad i = 1,2,\ldots,N \tag{5}$$

The problem thus becomes one of nonlinear programming. The general procedures developed to solve nonlinear programming problems are thus readily available and the mathematical basis not only lends elegance to the approach but also assures the attainment of the minimum of the function.

In a typical solution scheme for the preceding problem, the 'optimum' design variable vector v is evaluated as

$$\{V\}^{\nu+1} = \{V\}^{\nu} + \alpha\{S\} \tag{6}$$

where $\{V\}^{\nu+1}$ and $\{V\}^{\nu}$ = Design variable vectors in iteration $(\nu+1)$ and ν

$\{S\}$ = The direction of travel

α = A scalar variable that governs the move distance

However, one significant drawback of these methods is the rapid increase in computational effort with problem size. The approximation concepts suggested in Reference 6 (like design variable linkage, constraint deletion via regionalization and throw-away techniques) tend to alleviate this drawback.

Optimality Criteria Based Techniques:

In designs based on optimality criteria methods, an optimality criterion and a resizing formula are derived. The premise is used that when a design satisfying the optimality criterion is found, the objective function automatically takes on an optimum value.

The problem of weight minimization is stated as follows:

Minimize $$W(V) = \sum_{i=1}^{m} w_i V_i \tag{7}$$

subject to the constraint function that may be implicitly dependent on the design variables V_i

$$\Psi_j(V) = G_j(V) - G_{jo} \leq o \tag{8}$$

where G_{jo} = The desired value of $G_j(v_i)$ at some point o (j constraints j = 1,2,...,p)

Define W* as

$$W^* = W(V) + \sum_{j=1}^{p} \lambda_j [\Psi_j(V)] \tag{9}$$

where λ_j's = The Lagrange multipliers

Equating to zero the first partial derivatives of W* with respect to the design variables V_i

$$\frac{\partial W^*}{\partial V_i} = \frac{\partial W}{\partial V_i} + \sum_{j=1}^{p} \lambda_j \frac{\partial \Psi_j}{\partial V_i} = 0 \tag{10}$$

This equation will be simplified as:

$$\sum_{j=1}^{p} e_{ij} \lambda_j = 1 \qquad i = 1,2,\ldots,M \tag{11}$$

where, the elements of e_{ij} are given by

$$e_{ij} = \frac{\frac{\partial}{\partial V_i}[\Psi_j(V)]}{\frac{\partial}{\partial V_i}[W(V)]} \tag{12}$$

Equation (11) is called the optimality criterion.

Since the 'm' design variable unknowns of the problem are augmented by 'p' Lagrange multipliers, the problem has a total of (m+p) unknowns. Additional p equations are obtained by writing the original constraint conditions as follows:

$$\sum_{i=1}^{m} e_{ij}\rho_i v_i = G_{jo} \qquad j = 1,2,\ldots,p \tag{13}$$

where v_i = the volume of the element i

Equations (11) and (13) are nonlinear set of equations and they are solved by iterative methods. The difficulties involved in applying optimality criteria methods are basically associated with the need to identify the critical constraint set and the proper corresponding set of passive members. Nevertheless, the methods have been used in a large variety of problems with varying degrees of success (see Reference 8).

In what appears to be a merging of the two approaches discussed above, References 7, 9 and 10 discuss the benefits of 'dual' methods as promising trends to problems of structural synthesis.

USER-CONVENIENT TOOLS

Many of the ideas and equations presented in the preceding section were developed during the 1960's and early 1970's. Computer programs have also been developed and are available in the open literature to help the user perform structural optimization. An additional attractive feature of such development, from the viewpoint of the MSC/NASTRAN user community, is the availability in the open literature of what are known as 'black-box optimizers' based on mathematical programming techniques. It is the purpose of this paper to discuss the development of two user-convenient program products, CSAR/SIZING and CSAR/ØPTIM, that bring these optimization techniques within reach of the general MSC/NASTRAN user.

It must be emphasized in this context that attainment of the 'absolute minimum weight' design for large practical structures may neither be necessary nor computationally affordable. Rather, these techniques must be used as 'design improvement aids'.

CSAR/SIZING

This program performs the strengh sizing of axially stressed members (rods, bars, beams etc.) and biaxially stressed members (QUAD4, TRIA3, QUAD8, TRIA6, CSHEAR) based on fully stressed design. The premise of this design scheme is that every member is either fully stressed under at least one load condition or at minimum gage. It must be noted that there is no guarantee that such a scheme will always produce the 'minimum weight'. However, it is seen that the major improvements in weight reduction of the design occurs in the first few iterations of such a scheme and that a near optimum design can almost always be achieved. As such, the designers use the FSD procedure to obtain improvements of the original design by performing few iterations (say 3 to 5), to review the results and, perhaps, to proceed with another set of 3 to 5 iterations and so on, until a satisfactory design is obtained (see the flowchart of CSAR/SIZING shown in Figure 1). New bulk-data-type cards are introduced to supply the necessary information to the program to perform strength sizing. These are not MSC/NASTRAN bulk data cards but have a similar format since the users are familiar with such formats. These cards are shown in the Appendix and are self-explanatory.

<u>User Action</u>

User prepares the usual MSC/NASTRAN deck for static stress analysis and additionally provides the resizing information via the input data cards discussed in the Appendix. This single file called 'NASTSIZ' file is the only file the user has to provide to CSAR/SIZING. For example, the NASTSIZ file for the 10-bar truss example problem of Figure 2 is given in Table 1.

It may be observed from Table 1 that only 6 additional input cards are required to provide the information for resizing for the 10-bar truss example. Moreover, the data needed in those cards can be easily supplied by the average stress engineer.

CSAR/SIZING uses job control language cards and a software program consisting of subroutines SIZE1 and SIZE2F to obtain the design improvements. The TFSD type algorithm discussed in Equation 2 will also be provided as an option at a later date.

The results at the end of 5 iterations for the 10-bar truss (loading and constraints are given in Table 2) is shown in Table 3. Displacement constraints are not accounted for in Fully Stressed Design.

Notice the rapid decrease in weight in the first iteration (for large size problems involving many thousands of degrees of freedom, major weight decreases occur generally in the first few iterations). From an arbitrary set of starting areas for the rods, it was possible to obtain major design improvements in a few iterations.

CSAR/ØPTIM

The fully stressed design capability provided by CSAR/SIZING is easy to use and obtains an improved design in a few iterations. However, for problems involving displacement constraints and/or size constraints and for cases where the guarantee of the attainment of the minimum weight is a major goal of design, the mathematical elegance provided by mathematical programming methods is available to the user in the software product CSAR/ØPTIM. Two commonly used 'black-box type optimizers' for such purposes are the CØNMIN (Fortran program for Constrained Function Minimization - User's Manual: see References 12 and 13) and the NEWSUMT (Fortran Program for Inequality Constrained Function Minimization - User's Guide: see References 13 and 14) programs. CSAR Corporation has modified these programs, adapted them for use with MSC/NASTRAN and has developed the program CSAR/ØPTIM to provide users with the option to use either a constrained function minimization approach or a sequence of unconstrained minimization technique for optimization.

User Action

As for the case of CSAR/SIZING, the user prepares a file consisting of the usual MSC/NASTRAN deck for static stress analysis together with the cards for specifying optimization requirements. This file will be called a 'NASTØPT' file. The optimization requirements may be specified via the input cards shown in the Appendix and/or the MSC/NASTRAN bulk data cards for design sensitivity analysis. Even though the black-box optimizers work farily well for most applications, a knowledge of how the codes work, if available to the user, will undoubtedly be an asset while using CSAR/ØPTIM. Users are therefore encouraged to consult References 12, 13 and 14 to obtain an understanding of the minimization techniques.

The results at the end of 8 optimization loops for the 10-bar truss are shown in Tables 5 and 6. An optimization loop (see the flowchart of CSAR/ØPTIM shown in Figure 3.) is defined as one MSC/NASTRAN analysis, constraint evaluation, constraint deletion, MSC/NASTRAN sensitivity calculation, and one 'optimizer' processing (which processing may use a number of optimization surfaces with a number of one-dimensional minimizations in each surface as in the case of NEWSUMT or a number of iterations as in the case of CØNMIN). At this stage, there is weight savings of about 35% from the starting design and a feasible design is available. Proceeding further and performing more optimization loops, one can obtain the minimum weight. Reference 7 reports the minimum mass for this problem at 5060 lbs.

For large problems, the approximation concepts discussed in Reference 6 can and must be used to reduce the computational effort. The users can help by linking design variables (for example, all elements in the panel have same thickness or linearly varying thickness etc.); the program will help by using constraint deletion techniques. At the time of printing this paper, such techniques are being automated. The quality assurance testing is also in progress at this time. It is expected that some results on large problem solution will be presented at the conference.

CONCLUSION

Development of two software program products, CSAR/SIZING and CSAR/ØPTIM, for design improvements based on fully stressed design and structural optimization techniques is discussed in the paper. With the power of MSC/NASTRAN for large problem solution, these program products bring automated design improvement capability within easy reach of stress engineers. These programs can also be used as 'test beds' to test the suitability of other optimizer routines for various large-sized problems. The program products must be viewed as design improvement aids and, if used properly, may yield improved, efficient, and economical designs.

REFERENCES

1. Schmit, L.A.: Structural Design by Systematic Synthesis: Proceedings of the 2nd Conference on Electronic Computation ASCE, New York, 1960, pp. 105-122

2. Gellatly, R. A., and Gallagher, R. H.; and Luberacki, W.A.: Development of a Procedure for Automated Synthesis of Minimum Weight Structures, AFFDL-TDR-64-141, 1964.

3. Venkayya, V. B.: Design of Optimum Structures: Journal of Computers and Structures Vol.1 No. 1-2, 1971, pp. 265-309

4. Lansing, W.; Dwyer, W.; Emerton, R.; and Ranally, E.: Application of Fully-stressed Design Procedures to Wing and Empennage Structures: Journal of Aircraft, Vol. 8., No. 9, 1971, pp. 683-688

5. AGARD Conference Proceedings No. 123, 2nd Symposium on Structural Optimization, Milan, Italy, April 1973. (A number of excellent papers on structural optimization - Author's comment).

6. Lucien A. Schmit, Jr., and Hirokazu Miura: "Approximation Concepts for Efficient Structural Syntheseis" NASA CR-2552, March 1976

7. Fleury, Claude and Schmit, Jr., Lucien A., "Dual Methods and Approximation Concepts in Structural Synthesis" NASA Contractor Report 3226, December 1980.

8. Venkayya, V. B. "Structural Optimization: A Review and Some Recommendations" International Journal for Numerical Methods in Engineering, Vol. 13. pp. 203-228 (1978)

9. Fleury, C., "Structural Weight Optimization by Dual Methods of Convex Programming", International Journal for Numerical Methods in Engineering, Vol. 14, No. 12, 1979 pp. 1761-1783

10. Fleury C., and Sander G., "Relations between Optimality Criteria and Mathematical Programming in Structural Optimization: Proceedings Symposium on Applications of Computer Methods in Engineering, USC, Los Angeles, California, August 1977, pp. 507-520

11. Adelman, H. and Narayanaswami, R.: Resizing Procedure for Structures Under Combined Mechnaical and Thermal Loading: AIAA Journal, Vol. 14, No. 10, pp 1484-1486 October 1976

12. Garrett N. Vanderplaats: "CØNMIN - A Fortran Program For Constrained Function Minimization" NASA TM-X 62282 August 1973 (also CØSMIC Program # ARC-10836)

13. G. N. Vanderplaats and H. Miura: Short Course Text for Applications Optimization Methods in Engineering Design (with computer program handout) July 27-28, 1978

14. Hirokazu Miura and Lucien A. Schmit, Jr.,: "NEWSUMT - A Fortran Program for Inequality Constrained Function Minimization - Users Guide" NASA Contractor Report 159070 June 1979

Table 1. NASTSIZ File for 10-bar Truss
Example of Figure 2.

```
ID       TRUSS,SIZING
•
•
•
•
•
•
•
•
•        1. Usual MSC/NASTRAN data deck for SØL 24 or SØL 61 (if SØL 61, all
            elements to be resized must be in residual superelement)
•
•        2. All referenced MAT1 and MAT2 cards in the deck must have values
            for RHØ supplied by user.
•
•
•
•
•
•
•
•
ENDDATA
DESVAR,1,TRUSS,,CROD,A,,ALL
PRØPLIM,CROD,ALL,A,0.1
STRCNS,10,STRAXL,CROD,ALL,AXIAL,7,MAX
STRCNS,10,AXLMIN,CROD,ALL,AXIAL,7,MIN
STRLIM,7,MAXTEN,25000.,MAXCMP,25000.
ØPARM,17,FSD,5,0.05
```

Table 2. Definition of Problem 1: Planar 10-Bar Cantilever Truss (U.S. Customary Units).

Material:	Aluminum
Young's modulus:	$E = 10^7$ psi
Specific weight:	$\rho = 0.1$ lbm/in^3
Allowable stress:	$\sigma_a = \pm 25{,}000$ psi
Minimum area:	$D^{(L)} = 0.1$ in^2
Uniform initial area:	$D^{(o)} = 20.0$ in^2

Nodal Loading (1 load case)

Node	Load components (lbf)		
	X	Y	Z
2	0	-100,000	0
4	0	-100,000	0

Displacement Constraints

Node	Direction	Displacement limits (in)	
		Lower	Upper
1-4	Y	-2.0	+2.0

Table 3. Iteration History Data for First 5 Iterations for 10-bar Truss Shown in Figure 1. (no displacement constraints) CSAR/SIZING: Method = FSD

Member	Iteration 1		Iteration 2		Iteration 3		Iteration 4		Iteration 5	
	Area in^2	Stress ksi	Area in^2	Stress ksi	Area in^2	Stress ksi	Area in^2	Stress ksi	Area in^2	Stress ksi
1	20.0	9.768	7.815	23.815	7.444	24.322	7.242	25.052	7.257	25.417
2	20.0	2.006	1.605	18.912	1.214	19.081	0.927	19.945	0.740	20.490
3	20.0	-10.232	8.185	-26.132	8.556	-25.589	8.759	-24.954	8.743	-24.654
4	20.0	-2.994	2.395	-29.080	2.785	-27.588	3.103	-26.268	3.260	-26.023
5	20.0	1.774	1.419	11.595	0.658	6.416	0.169	0.482	0.1	3.897
6	20.0	2.006	1.605	18.912	1.214	19.083	0.927	19.947	0.740	20.491
7	20.0	7.399	5.919	27.211	6.443	26.109	6.728	24.922	6.707	24.362
8	20.0	-6.743	5.395	-22.573	4.871	-23.535	4.585	-25.114	4.606	-25.929
9	20.0	4.234	3.387	29.080	3.940	27.579	4.346	26.524	4.611	26.023
10	20.0	-2.837	2.270	-18.911	1.717	-19.083	1.311	-19.953	1.046	-20.492
Mass	8393		1693		1651		1591		1614	

Table 4. NASTØPT File for 10-bar Truss
Example of Figure 2.

```
ID        TRUSS,SIZING
•
•
•
•
•
•
•
•
•        1. Usual MSC/NASTRAN data deck for SØL 24 or SØL 61 (if SØL 61, all
            elements to be resized must be in residual superelement)
•
•        2. All referenced MAT1 and MAT2 cards in the deck must have values
            for RHØ supplied by user.
•
•
•
•
•
•
•
•
ENDDATA
DESVAR,1,TRUSS,,CROD,A,,ALL
PRØPLIM,CROD,ALL,A,0.1
STRCNS,10,STRAXL,CROD,ALL,AXIAL,7,MAX
STRCNS,10,AXLMIN,CROD,ALL,AXIAL,7,MIN
STRLIM,7,MAXTEN,25000.,MAXCMP,25000.
DISCNS,29,DISP1,ALL,T2,2.0,MAX
DISCNS,29,DISPMIN,ALL,T2,-2.0,MIN
ØPARM,17,ØPTIM1,20
```

Table 5. Iteration History Data for First 8 Iterations.
CSAR/ØPTIM: Method = ØPTIM1; Optimizer = CØNMIN

Member	Area of Truss Members							
	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6	Iter 7	Iter 8
1	20.0	32.2	31.24	31.85	33.14	30.61	31.36	31.74
2	20.0	7.92	6.35	5.18	4.28	3.16	2.72	2.05
3	20.0	32.7	32.47	32.87	31.82	27.01	26.34	25.89
4	20.0	10.8	12.31	13.49	14.34	13.55	14.14	14.68
5	20.0	2.18	1.44	1.03	0.75	0.54	0.45	0.32
6	20.0	7.92	6.35	5.18	4.28	3.09	2.40	1.64
7	20.0	18.80	18.80	17.63	16.11	12.98	11.78	10.84
8	20.0	18.2	17.93	18.79	19.87	18.98	19.80	20.84
9	20.0	11.2	17.96	18.77	19.73	18.99	20.01	20.92
10	20.0	11.2	8.80	7.04	5.76	4.11	3.43	2.52
Total Mass	8393	6597	6480	6395	6319	5610	5588	5554

Minimum mass reported in Reference 7 = 5060

Table 6. Iteration History Data for First 8 Iterations.
CSAR/ØPTIM: Method = ØPTIM2; Optimizer = NEWSUMT

Member	Area of Truss Members							
	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6	Iter 7	Iter 8
1	20.0	30.0	28.92	29.7	29.6	30.4	30.96	31.37
2	20.0	7.46	5.46	4.47	4.44	3.34	2.365	1.488
3	20.0	30.72	29.62	29.5	29.4	28.17	26.94	25.78
4	20.0	10.46	11.32	12.7	12.6	13.38	14.13	14.74
5	20.0	1.08	0.10	0.17	0.149	0.128	0.129	0.125
6	20.0	6.67	4.84	4.14	4.17	3.057	2.05	1.09
7	20.0	17.98	17.08	15.7	15.6	13.78	11.99	10.31
8	20.0	17.12	15.85	17.9	17.9	18.91	19.98	20.85
9	20.0	14.19	15.64	18.1	18.0	18.94	19.99	20.85
10	20.0	10.54	7.71	6.32	6.28	4.73	3.35	2.11
Total Mass	8393	6145	5755	5892	5834	5694	5572	5442

Minimum mass reported in Reference 7 = 5060

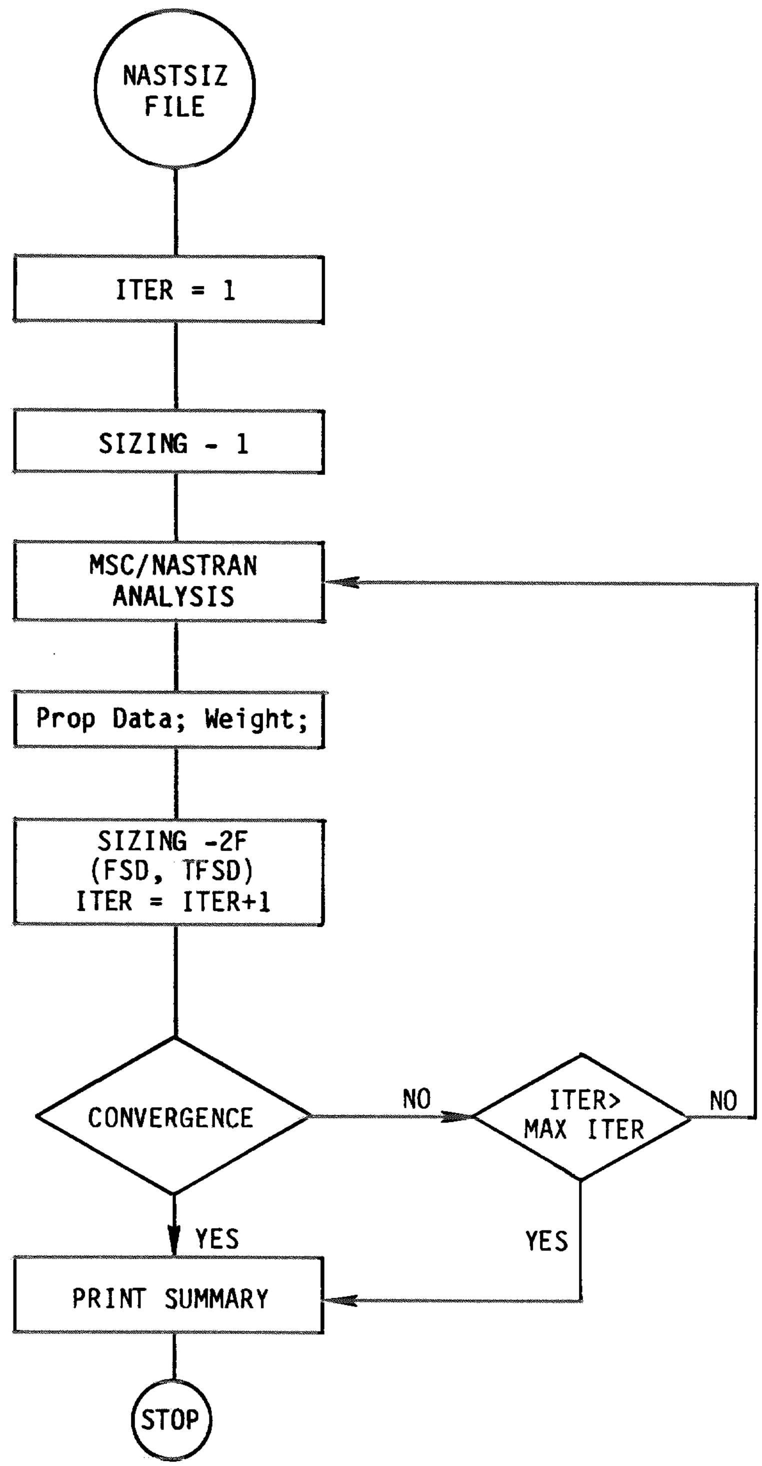

Figure 1. CSAR/SIZING FLOWCHART.

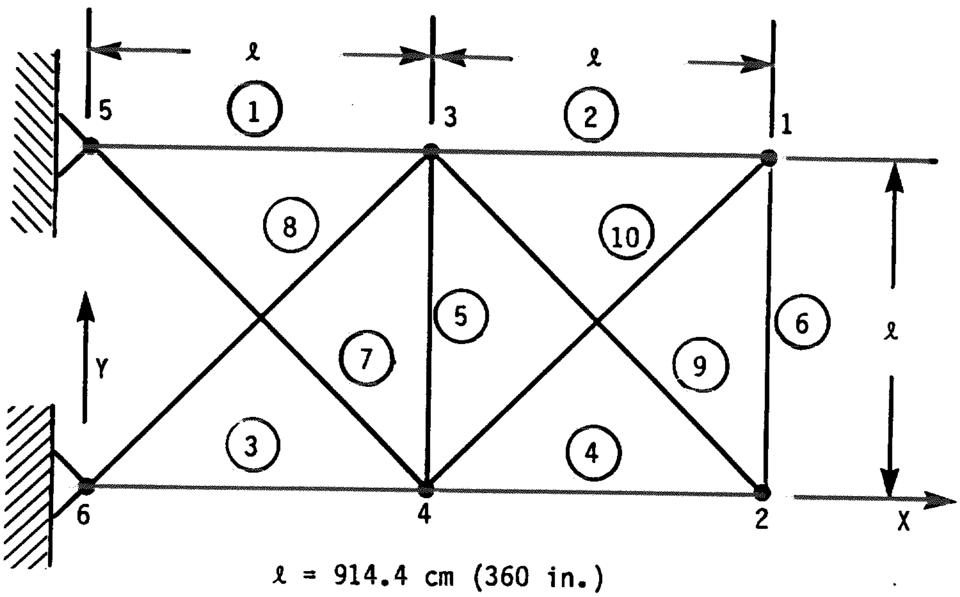

Figure 2. Planar Ten-bar Truss.

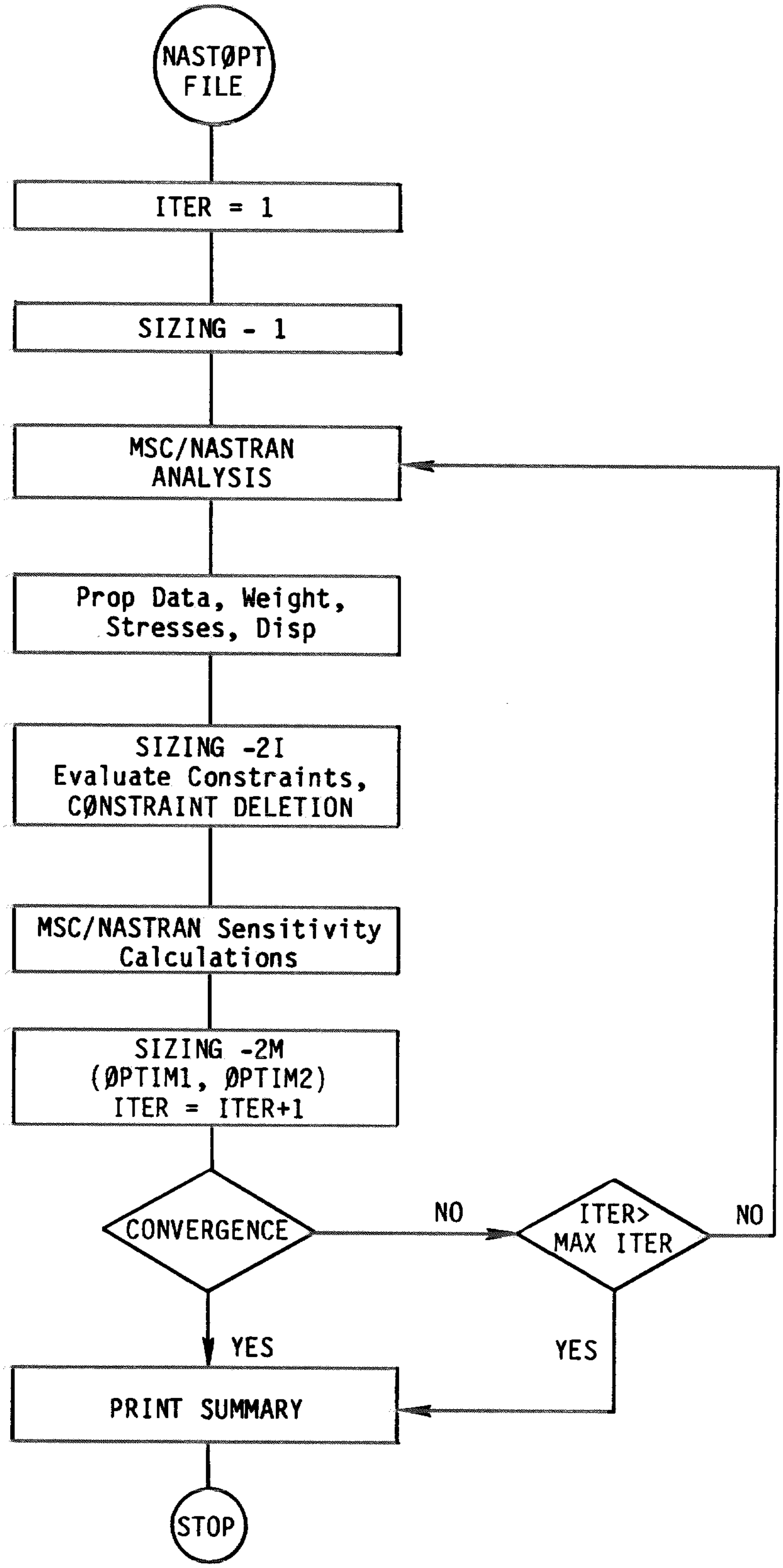

Figure 3. CSAR/ØPTIM FLOWCHART.

Appendix

Input Data Card <u>DESVAR</u> Design Variable

Description: Defines a Design Variable for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
DESVAR	ID	LABEL	DELTAB	ELTYPE	PLIST	ALPHA	ELIST	VARSTAT	
DESVAR	1	PRØB1		CRØD	A		ALL	UNIQUE	

Field	Contents
ID	Design Variable ID (Integer)
LABEL	Label used to describe variable in output (BCD)
DELTAB	Change ΔB in the dimensionless design Variable, B, to be used (Real) (Default = 0.02)
ELTYPE	CRØD, CONRØD, CBAR, CBEAM, CQUAD4, CTRIA3, CQUAD8, CTRIA6, CSHEAR (BCD)
PLIST	Property List (BCD or Integer) A,T or Integer, if Interger refers to PLIST card (PLIST card not presently supported.)
ALPHA	Exponent of actual element property versus design variable (Real $\neq$ 0.0) (Default = 1.0)
ELIST	ALL or Integer (BCD or Integer) If Integer, refers to ELIST card
VARSTAT	Variable Status (BCD or Integer) BCD : Unique : Distinct (Default) Same : Same for all elements Integer : If Integer, refers to RELID (Relationship Identifier) card This option presently not available.

Input Data Card <u>DISCNS</u> Displacement Constraint

Description: Defines displacement constraints for use with optimization techniques of CSAR/ØPTIM software product

Format and Example

1	2	3	4	5	6	7	8	9	10
DISCNS	DISCID	LABEL	GRLIST	CØMP	MINLIM	MAXLIM			
DISCNS	29	DISP1	ALL	T3	-2.0	+2.0			

Field	Contents
ID	Displacement Constraint ID (Integer)
LABEL	Label used to describe the constraint in output (BCD)
GRLIST	ALL (BCD) or Integer; If integer, refers to GRLIST card
CØMP	Component to be constrained T1, T2, T3, R1, R2, R3 (BCD)
MINLIM	Minimum value of constraint (Real or blank)
MAXLIM	Maximum value of constraint (Real or blank)

Remarks

If MINLIM or MAXLIM field is blank, such constraint will not be applied.

Input Data Card <u>ELIST1</u> Element list

<u>Description</u>: Defines a list of element IDs for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

<u>Format and Example</u>

1	2	3	4	5	6	7	8	9	10
ELIST1	ID	EID1	EID2	EID3	EID4	EID5	EID6	EID7	
ELIST1	7	1	5	9	14				

<u>Field</u>	<u>Contents</u>
ID	ELIST1 ID (Integer)
EID1, EID2, ..., EIDn	Element IDs (Integer)

<u>Alternate form 1</u>

ELIST1	ID	EID1							
ELIST1	ID	ALL							

<u>Alternate form 2</u>

ELIST1	ID	EID1	EID2	EID3	EID4				
ELIST1	ID	EID1	THRU	EID2					

Input Data Card ELISTZ Element list form 2

Description: Defines a list of ELIST2 IDs for use with FSD, Optimization technique of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
ELIST2	ID	EL-1	EL-2	EL-3	EL-4	EL-5	EL-6	EL-7	
ELIST2	27	17	19	21					

Field	Contents
ID	ELIST2 ID (Integer)
EL-1, EL-2, ..., EL-n	ELIST2 IDs (Integer)

Alternate form 1

ELIST2	ID	EL-1	EL-2	EL-3					
ELIST2	ID	15	EXCEPT	16					

Input Data Card <u>GRLIST</u> Grid list

Description: Defines a list of grid IDs for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
GRLIST	ID	GRID1	GRID2	GRID3	GRID4	GRID5	GRID6	GRID7	
GRLIST	7	1	5	9	14				

Field	Contents
ID	GRLIST ID (Integer)
GRID1, GRID2, ..., GRIDn	Grid IDs (Integer)

Alternate form 1

GRLIST	ID	GRID1							
GRLIST	ID	ALL							

Alternate form 2

GRLIST	ID	GRID1	GRID2	GRID3	GRID4				
GRLIST	ID	GRID1	THRU	GRIDn					

Input Data Card <u>ØPARM</u> Optimization Parameter

Description: Defines parameters for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
ØPARM	ID	METHØD	ITMAX	TØLER					
ØPARM	17	FSD	3	0.05					

Field	Contents
ID	Optimization Parameter ID (Interger)
METHØD	Method used: FSD, ØPTIM1, ØPTIM2 (BCD)
ITMAX	Maximum Iterations (Integer) (Default = 5)
TØLER	Tolerance (Real)

Remarks:

The iteration scheme will be terminated when (i) total number of iteration exceeds ITMAX or (ii) $\frac{M_i - M_{i-1}}{M_i}$ < TØLER and the design is feasible: (M_i is the mass of Iteration i; $M_0 = 0$)

Input Data Card PRØPLIM Property limit

Description: Defines limiting values of property data for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
PRØPLIM	ELTYPE	ELIST	VARIABLE	MINVAL	MAXVAL				
PRØPLIM	CRØD	ALL	A	0.1	2.0				

Field	Contents
ELTYPE	CRØD, CONRØD, CBAR, CBEAM, CQUAD4, CTRIA3, CQUAD8, CTRIA6, CSHEAR (BCD)
ELIST	All or Integer (BCD or Integer) If integer, refers to ELIST card
VARIABLE	A,T (BCD)
MINVAL	Minimum Value (X^L) (Real)
MAXVAL	Maximum Value (X^u) (Real)

Note: Variable $X^L \leq X \leq X^u$

Input Data Card <u>STRCNS</u> Stress constraint

Description: Defines Stress constraints for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
STRCNS	ID	LABEL	ELTYPE	ELIST	STRTYP	STRLIM	ØPT		
STRCNS	10	STRI	CRØD	ALL	AXIAL	7	MAX		

Field	Contents
ID	Stress constraint ID (Integer)
LABEL	Label used to describe the constraint in output (BCD)
ELTYPE	Element type (BCD) CRØD, CØNRØD, CBAR, CBEAM, DQUAD4, CTRIA3, CQUAD8, CTRIA6, CSHEAR
STRTYP	Stress Type (BCD) AXIAL, TORSION, (Rod element) AXIAL, MOMENT1, MOMENT2, TØRQUE (Bar, Beam) AVSHR (Shear element) MEMHVM, BENHVM, MAXPRNM, MAXPRNB (Plate element)
STRLIM	Stress limit ID (Integer)
ØPT	Constraint equation option MAX, MIN (BCD) (Default = MAX)

Input Data Card STRLIM Stress limit

Description: Defines Stress allowables for use with FSD, Optimization techniques of CSAR/SIZING and CSAR/ØPTIM software products

Format and Example

1	2	3	4	5	6	7	8	9	10
STRLIM	ID	ITEM1	VALUE	ITEM2	VALUE	ITEM3	VALUE		
STRLIM	7	MAXTEN	20000.	MAXCMP	20000.	MAXSHR	11546		

Field	Contents
ID	Stress Limit ID (Integer)
ITEM1, ITEM2, ITEM3	Any of the following keywords (BCD) MAXTEN, MAXCMP, MAXSHR MAXTEN1, MAXTEN2 MAXCMP1, MAXCMP2
VALUE	Value for the particular item (Real)

DESIGN IMPROVEMENTS USING DESIGN SENSITIVITY ANALYSIS CAPABILITY OF MSC/NASTRAN

by

R. Narayanaswami
President

Computerized Structural Analysis & Research
(CSAR) Corporation

Northridge, California
(818) 368-8625

- Engineers Are Interested in a Capability to Obtain Design Improvements in an Automated Fashion
- Minimum Weight Design: Find Minimum Weight Design of Structural Model
- Structural Optimization: Optimize the Objective Function
- Structural Synthesis: As Above
- We Use the Term "Design Improvements" to Refer to the Process of Obtaining the Minimum Weight Design

DESIGN IMPROVEMENTS

- Fully Stressed Design (FSD)
- Thermal Fully Stressed Design (TFSD)

FULLY STRESSED DESIGN (FSD)

- Intuitive Feeling

- Easy to Understand and use

$$A_{i+1} = \frac{\sigma_i}{\sigma_a} A_i \quad \text{(Resizing Algorithm for Rod Element)}$$

- Very Useful for Preliminary Designs

FULLY STRESSED DESIGN (Cont.)

Limitations:

- No Guarantee that will Always Lead to a Minimum Weight Design
- Displacement Constraints are not Considered

Strengths:

- Simplicity, Ease of use and (Computational) Cost Savings

THERMAL FULLY STRESSED DESIGN (TFSD)

- FSD Converges Slowly if Thermal Stresses are Present

  Reason: Thermal Stresses are Insensitive to Resizing ($E \propto T$)

- Adelman and Narayanaswami (AIAA Journal October 1976 pp. 1484–1486) Proposed TFSD

$$A_{i+1} = \frac{\sigma_{M_i}}{\sigma_{a,M} - \sigma_{T_i}} A_i$$

- More Useful Than FSD if Thermal Stresses Dominate

DESIGN IMPROVEMENTS

Mathematical Programming Methods

- Constrained Function Minimization
- Sequence of Unconstrained Minimization Techniques (SUMT)

Optimality Criteria Methods

- Develop Optimality Criterion and Resizing Algorithm
- Such Procedures Have Been Developed for Displacement, Vibration Frequency, Flutter, Temperature Constraints
- FSD and TFSD Techniques

MATHEMATICAL PROGRAMMING TECHNIQUES

- Minimize an Objective Function

$$F(\bar{x}) = \sum_{i=1}^{N} W_i X_i$$

Subject to a Given Set of Constraints

$$g_q(\vec{x}) \leq o \qquad q = 1, 2, .., Q$$

and/or Side Constraints

$$X_i^{(L)} \leq X_i \leq X_i^{(U)} \qquad i = 1, 2, ..., N$$

MATHEMATICAL PROGRAMMING TECHNIQUES (Cont.)

- Typical Solution

$$\{X\}^{\nu+1} = \{X\}^{\nu} + \alpha\{S\}$$

α : Move Distance $\{S\}$: Direction of Travel

- Constrained Function Minimization
- Convert Constrained Function Minimization into Sequence of Unconstrained Function Minimizations by Introducing Penalties for Constraint Violations
- Black–Box Optimizers: CØNMIN & NEWSUMT

MATHEMATICAL PROGRAMMING TECHNIQUES (Cont.)

<u>Strengths</u>

- Mathematical Elegance
- Guarantee of Obtaining 'OPTIMUM' (Local: Yes
Global: Maybe)

<u>Limitations</u>

- Computational Costs

<u>Remedial Measures</u>

- Design Variable Linking
- Use of Reciprocal Variables After Linking
- Constraint Deletion (Regionalization, Throw–Away Concepts)
- Approximate Problem Solution – Taylor Series Expansion

MATHEMATICAL PROGRAMMING METHODS

Black–Box Optimizers

- CØNMIN (Constrained Function Minimization)
- SUMT (Sequence of Unconstrained Minimization Technique)
  - Extended Penalty Function

$$\Phi(\vec{X}, r_p) = F(\vec{X}) + r_p \left[\sum_{q=1}^{Q} H_q(\vec{X}) + \sum_{j=1}^{NDV} (L(X_j) + U(X_j)) \right]$$

- Minimize Φ for a Sequence of Decreasing Values of r_p

OPTIMALITY CRITERIA TECHNIQUES

- Problem Formulation Same as Before

$$F^* = F + \sum_{j=1}^{p} \lambda_j \, \psi_j \, (X)$$

Where λ's are Lagrange Multipliers

- An Optimality Criterion and a Resizing Algorithm are Developed. Iterative Scheme is used to Solve the Resulting Equations.

OPTIMALITY CRITERIA TECHNIQUES (Cont.)

Strengths

- Computational Ease and Savings

Limitations

- Need to Know Which Constraints are Active

Usage

- Optimality Criteria Have Been Developed For Strength, Displacement, Buckling, Vibration Frequency, Temperature etc.

DESIGN IMPROVEMENTS USING MSC/NASTRAN

- All Such Techniques for Minimization of Objective Functions Require Evaluation of Sensitivity Coefficients (Gradient of Constraints with Respect to Design Variables)
- MSC/NASTRAN Version 63 has Design Sensitivity Analysis Capability
- CSAR Corporation has Developed Two User Convenient Products, CSAR/SIZING and CSAR/ØPTIM

CSAR/SIZING

- Fully Stressed Design
- Easy User–Convenient Input
- Uses JCL and Software Program
- Major Weight Decreases Occur in First Few Iterations in FSD. Users Try 3 to 5 FSD Iterations, Review Results and Proceed Again if Necessary.

CSAR/ØPTIM

- Uses JCL and Software Program
- Provides Option of Using Either Constrained Function Minimization (CØNMIN) or Sequence of Unconstrained Function Minimization Techniques (NEWSUMT)
- Easy User Input
- Brings Mathematical Programming Techniques Within Easy Reach of Average Stress Engineer

CONCLUSION

- MSC/NASTRAN Version 63 Has Good Design Sensitivity Analysis Capability
- CSAR/SIZING and CSAR/OPTIM Software Products Bring the Capability of Minimum Weight Design Within Easy Reach of Average Stress Engineer
- If Used Properly, Such Capability May Yield Improved, Efficient, Economical Designs