STRUCTURAL SYNTHESIS USING MSC/NASTRAN

G. N. Vanderplaats*, H. Miura*, G. Nagendra** and D. Wallerstein**

ABSTRACT

An automated structural synthesis capability has been developed as a set of internal modules of MSC/NASTRAN. The program takes advantage of utilities available in MSC/NASTRAN and is closely coupled with the sensitivity analysis capability to achieve overall numerical efficiency. The program architecture and data structure are designed open-ended to accommodate future extensions as new technologies become available. The concept of a design model is established as contrasted with the concept of analysis model, and the user interface to represent the design model is defined. In order to enhance flexibility in defining a variety of design models, a new capability to read mathematical equations as a form of bulk data is introduced. It is used to prescribe the relations between design variables and analysis model parameters as well as to transform structural responses into special forms that the user may desire for the design model definition. Currently, the synthesis capability is limited to sizing problems, but the data and program structures are designed to readily accommodate shape design in the near future.

1. INTRODUCTION

In 1983, the design sensitivity analysis (DSA) capability was installed in MSC/NASTRAN, Version 63. This technique was well known since the 1960's, but none of the commercially available finite element structural analysis codes had this capability before MSC/NASTRAN. As it turned out, implementation in MSC/NASTRAN was an epoch making success and opened up a new era of practical applications of structural design optimization. This is because the other building block, numerical optimization software, was already available [1-2], and the basic strategy to assemble a complete system efficiently had been developed by the mid-1970's [3]. Consequently, several codes which coupled optimization with MSC/NASTRAN appeared almost immediately after the release of of the design sensitivity capability [4-8], and there seem to be a number of similar development projects within company boundaries yet to be announced outside.

However, building a design synthesis capability external to MSC/NASTRAN requires 1-3 man years and must live with various limitations, because direct access to the MSC/NASTRAN utilities and database is not readily available. For example, if analysis bulk data is prepared taking advantage of any data

---

* Engineering Design Optimization, Inc., Santa Barbara, California

** The MacNeal-Schwendler Corporation, Los Angeles, California

generation capability of MSC/NASTRAN, it has to run through the IFP and obtain a complete sorted card image file. Also analysis capabilities, especially DSA, are closely coupled with a part of the design optimization capabilities, therefore if they are not coordinated, excessive overlap in data processing may occur for certain types of problems. Furthermore, recent studies indicate that some of the capabilities embedded in the finite element analysis may have to be used in the approximate analysis to improve the overall efficiency. In this context, it is important for the finite element analysis and approximate analysis capabilities to share certain program modules and data, and so a high level of integration is recognized to be a matter of critical importance.

Development of the internal MSC/NASTRAN structural optimization capabilities was started in 1986 as a joint effort of Engineering Design Optimization (EDO) and MacNeal-Schwendler Corporation (MSC), with funding provided by MSC. From the beginning, we were well aware that structural optimization is not a fully mature technology and that we must be prepared to make necessary adjustments in the future with regard to implementation of new capabilities. However, MSC/NASTRAN provides a solid foundation to build a flexible framework that should withstand a changing technical environment in the foreseeable future. Final evaluation of the performance of this capability will be made by the users, but we are confident that the capabilities we are preparing will be useful in a practical design environment and accepted favorably by the user community.

2. DESIGN OF THE STRUCTURAL SYNTHESIS SYSTEM AND USER INTERFACE

To develop the needed capability, the basic principles we set during the planning phase were:

(1) The concept of a **design model** must be established as contrasted with the concept of an **analysis model**.
(2) The definition of a design model must be flexible and adaptive to changing demands and requirements.
(3) The measure of computational efficiency is the number of complete finite element analyses including sensitivity.
(4) Recognizing that technology is making steady progress, the program and the data structure must be open ended.

In Fig. 1, a conceptual diagram represents the relation between conventional analysis and the new design synthesis capabilities, with emphasis on the data structure representation. The concept described in this figure is general and flexible to accommodate our basic strategic plans.

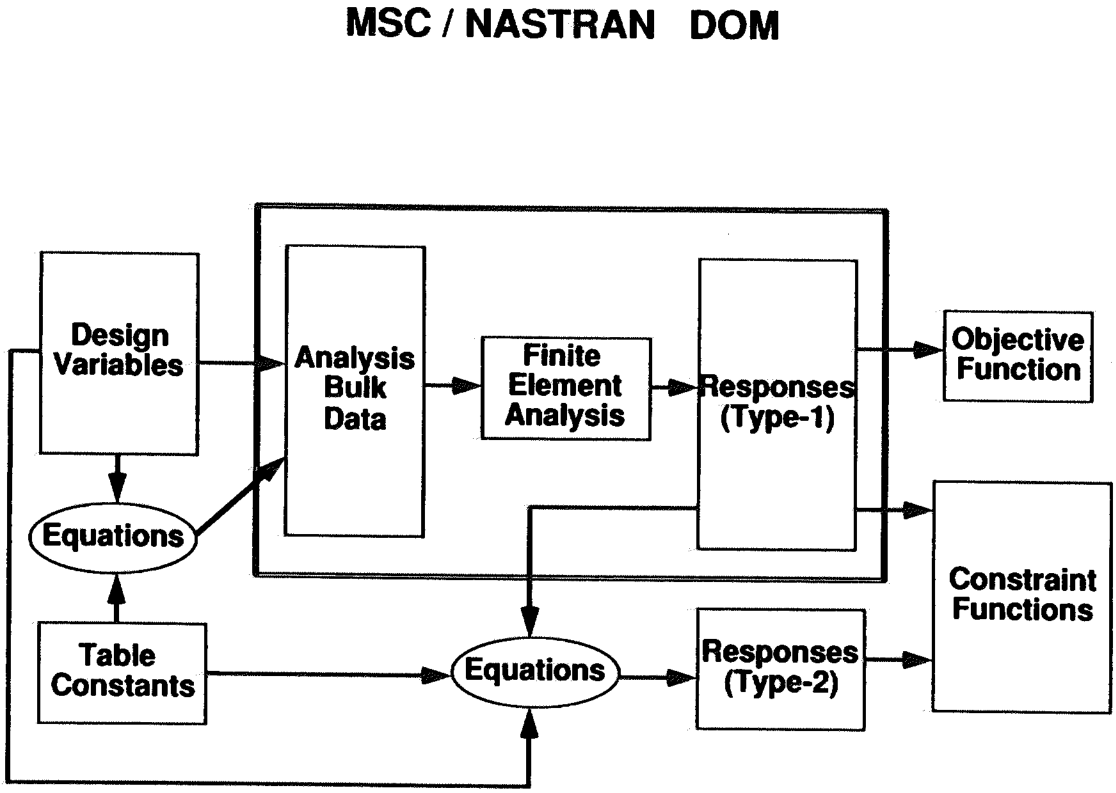

FIG. 1 Relation of Analysis and Synthesis Capabilities

2.1 Definition of Design Problems

One of the reasons that MSC/NASTRAN has become an overwhelmingly popular structural analysis program is that MSC established the procedures to quickly respond to the changing user demands for analysis capabilities. With regard to design problems, flexibility requested by the users is expected to be far more demanding compared to the need for flexibility to represent analysis models. In the aerospace industry, for example, finite element methods are often not used to generate stress data for design, instead they are used to find internal force distributions. Decisions on detailed sizes depend on variety of requirements and each company may have different procedures and design charts to decide materials, rivet spacings, stiffener shapes, etc. Similarly, design methods for modern composite materials are changing very rapidly. It is our conclusion that the data structure that satisfies all the design requirements is probably impossible to find. Even if we can satisfy all requirements today, innovative designs of tomorrow may make our data structure obsolete. On the other hand, if the user is requested to write programs in a high level language (such as FORTRAN or DMAP), a large number of users who are extremely busy handling day-to-day tasks may be excluded from the use of the automated synthesis capability.

We expended considerable effort to find an acceptable answer to this problem. For the first release, our answer is the capability to provide analytic equations that can be given as a part of bulk data. This is new to MSC/NASTRAN and at this time is restrictive compared to the FUNCTION statement of FORTRAN, but imaginative users will find it convenient to define a variety of specific requirements. The arguments of each equation may be design variables, constants from the constant table, and quantities obtained by the MSC/NASTRAN analysis, such as displacements, stresses, frequencies, etc. For example, a condition to avoid local crippling for a straight pipe subject to compressive axial load may be expressed as:

$$0.4\ E\ t\ /\ D - S_c \geq 0 \tag{1}$$

where the pipe diameter ,D, and thickness ,t, are design variables, the elastic modulus ,E, is a constant and compressive stress S_c is computed by finite element analysis. A design condition like this can be introduced to the design problem, using the equation capability, without any compromise or approximation.

In order to describe design problems, 13 new bulk data formats are added as shown below.

DESVAR : Design variables
DLINK : Define dependent design variables
DVPREL1: Linear relation between analysis model parameters and design variables
DVPREL2: Nonlinear relation between analysis model parameters and design variables through user supplied analytic equations
DVPREL3: Relation between Beam/Bar section properties and design variables through a standard cross-section library
DEQATN : User supplied equations
DTABLE : Table of constants
DRESP1 : Structural responses that are directly computed by MSC/NASTRAN analysis : (called Type-1 Responses)
DRESP2 : Synthesized responses by means of user supplied equations as functions of design variables, constants, and Type-1 responses. (called Type-2 Responses)
DESOBJ : Selection of the objective function.
DCONSTR: Define design constraints.
DSCREEN: Parameters to control the constraint screening process.
DOPTPRM: Parameters to control the overall design process.

We expect that the method of describing design problems may have to be extended in the future. There are several options considered at this time, however, we need feedback from the users as to the desirable directions.

In the immediate future, the bulk data given above have to be prepared manually or by custom made programs, since no preprocessors are available yet to generate these data. Similarly, synthesis results are available in the form of line printer output only. We anticipate pre- and post-processors that can handle design problem descriptions will become available in the near future.

2.2 Concept of a Design Model

A design model is defined by design variables, design constraints and a design objective. Since the decision tools used here are nonlinear programming algorithms, a design problem is formulated in the following form:

Select design variables $X = [x_1, x_2, \ldots x_{NDV}]$ so that a scalar function $F(X)$ is minimized, while satisfying following inequality constraints imposed on scalar functions $G_j(X)$;

$$G_j(X) \leq 0.0 \qquad j = 1, 2, \ldots J \tag{2}$$

Here we make a strong distinction between the design model and the analysis model. The analysis model is defined by member properties, joint coordinates, etc. and the results of the analysis are called responses, including displacements, stresses, and eigenvalues as examples. On the other hand, while we may wish to treat properties, such as member thicknesses, as design variables, the design model is much broader than that. The design variables can be any convenient representation, such as a thickness distribution, while the analysis variables are dependent functions of these. Similarly, it is not necessary to minimize or maximize, or constrain, only responses which are directly provided by the finite element analysis, although this is certainly allowed. The user may input equations defining "composite" functions which are related to the design variables and the responses. For example, it is possible to minimize some combination of displacements if there is some reason to do so. Also, the local crippling constraint given in equation 1 may be included in this fashion. Thus, through the design model concept, the user has considerable flexibility to create his own functions to be used in optimization. The only requirement is that these functions themselves be related to the design variables and basic responses for which MSC/NASTRAN can calculate sensitivity. Then, when the sensitivities of these composite functions are required during optimization, they are calculated internally from the available information.

In order to complete the definition of structural design problems, the relations between design model and structural analysis model must be defined.

2.2.1 Selection of design variables

Finite element structural analysis models are a faithful representation of actual structures compared to models used in the pre-finite element era. However, the ultimate purpose of structural analysis is to capture structural responses to the accuracy demanded by the analysis engineers, hence they don't hesitate to use simplifications and abstractions of the models as long as they can obtain the desirable accuracy in predicting responses that they are interested in. For example, in the analysis models, a coil spring may be modeled as a rod element and characteristics of a beam cross section are represented by the secondary quantities such as section area and moment of inertia instead of actual dimensions that appear in drawings.

In the design model, it is possible to use these secondary quantities as design variables, but they are, in general, mutually dependent quantities. Therefore, if their relations are not incorporated properly, the synthesis process may yield cross sections that are physically unattainable. In this context, it is necessary to separate design variables completely from parameters used in the analysis models.

In practice, adequate selection of design variables is critically important to successful applications of an automated synthesis capability. In general, it is extremely unlikely that each of a large number of finite elements is associated with distinct design variables. Therefore, selection of design variables is usually based on dimensions of actual hardware and a group of elements associated with each design variable must be identified by going back to the model generation process.

For some applications, it may be convenient to introduce linear relations among design variables, which can be provided by means of DLINK cards. This concept is similar to the multi-point constraints applied to displacement degrees of freedom in the analysis. Relations introduced by a set of DLINK cards will yield the relation given as:

$$X = \left|\begin{array}{c} X_I \\ \hline X_D \end{array}\right| = \left|\begin{array}{c} 0 \\ \hline C_D \end{array}\right| + \left|\begin{array}{c} I \\ \hline T_1 \end{array}\right| * X_I \qquad (3)$$

where X_I and X_D are independent and dependent design variables, respectively. C_D is a constant vector and T_1 is a transformation matrix.

Theoretically, the maximum number of design variables is limited only by the size of available memory, but it seldom exceeds 200. For large numbers of design variables, such as over 500, we don't have sufficient experience to find where the the theoretical or computational weaknesses, if any, are.

One group of the relations between design models and analysis models are the relations between the design variables and the parameters used in the description of analysis models. The current sensitivity analysis data structure permits the user to define design variables by DVAR cards and their relations to the analysis model parameters by DVSET cards. The relations between them are always defined in the form:

$$P_{NEW} = P_o + P_{REF} \left(B^a - 1 \right) \qquad (4)$$

where P_o and P_{REF} are original and reference values of analysis model parameters, B is a normalized independent design variable, 'a' is the power factor and P_{NEW} is the new value of analysis model parameter. This form is sufficiently general for sensitivity analysis, where only infinitesimal changes are considered. For design synthesis, it is necessary to consider finite changes, and the nonlinearity associated with design variable-property relations must be preserved as accurately as possible, therefore the form of relation given in equation (4) is not sufficient.

The bulk data cards used to describe design variables and analysis model parameters are the DVPREL1, DVPREL2 and DVPREL3 cards. DVPREL1 cards describe analysis model parameters as the <u>linear</u> functions of design variables as follows:

$$P_1 = P_{1o} + [T_2]*X_I \qquad (5)$$

where P_1 is the vector of analysis model parameters of this type, and P_{1o} and T_2 are a constant vector and matrix, respectively. Note that the most primitive one-to-one relations are included as a special case of equation (4), with $P_{1o} = 0$ and $T_2 = I$. Yet it can also represent the generalized reduced basis concepts.

DVPREL2 cards are used to describe analysis model parameters as nonlinear functions of design variables. Each of these cards refers to an equation provided by the user and a list of design variables and constants that are the arguments of the equation.

DVPREL3 cards are convenient to describe standard beam sections in the design model. Currently there are nine standard cross sections in the section library. For these sections, the user does not have to write equations to compute analysis model parameters from the section dimensions. Instead, the section ID and the choice of dimensions to be considered as design variables are prescribed in DVPREL3 cards.

DVPREL2 and DVPREL3 cards allow the user to design the physical dimensions of an element, even though the analysis and sensitivity information is based on section properties.

2.2.2 Selection of design objective and constraints

As described in the problem formulation in (2), the user is requested to identify with one scalar function to be minimized (or maximized) as the measure to rank the proposed designs. The design constraints must also be formulated as inequalities imposed on a group of scalar functions. The objective functions F(X) and constraint functions $G_i(X)$ are usually implicit functions of design variables, X. Explicit arguments prescribed in these functions are usually structural responses, which are computed by the finite element analysis. Although MSC/NASTRAN can calculate a large number of types of structural responses, those that can be used with structural design synthesis are the following:

(1) Structural weight
(2) Structural volume
(3) Displacements
(4) Stresses
(5) Strains
(6) Internal forces
(7) Stresses in composite materials
(8) Strains in composite materials
(9) Failure criteria associated with composite materials
(10) Natural frequencies (actually, the associated eigenvalues)
(11) Buckling load factor

The list will certainly grow in the future, in concert with the development of the sensitivity analysis capability.

The user must designate all structural responses that are computed directly by MSC/NASTRAN analysis and also used in the design problem (direct, or type-1 responses). This task is carried out by preparing DRESP1 cards. In order to save the user effort preparing data, it is not necessary to prepare a DRESP1 card for each stress component for all elements of interest. DRESP1 cards for stress refer to property cards instead of element cards. Then the stresses in all elements that refer the specific property cards are automatically included in the list of direct responses. If the nature of the design problem calls for more complex quantities to describe objective and/or constraint functions, synthetic responses, or type-2 responses, may be prescribed by DRESP2 cards. A DRESP2 card makes reference to a user supplied equation, and arguments to the equation may include design variables, constants and direct (type-1) responses. Within one equation, the user is not allowed to mix different types of responses or responses from a multiple number of subcases.

An objective function is selected by a DESOBJ card, that designates one of DRESP1 or DRESP2 cards, and specifies whether that response must be minimized or maximized. Obviously, a DESOBJ card is not allowed to make reference to a response card that generates a multiple number of responses. Note, however, that any response (type-1 or type-2) that is allowed in optimization may be the objective function.

Constraints are described by DCONSTR cards. Each of DCONSTR card refers to a DRESP1 or DRESP2 card together with the lower and upper bound data to be imposed on the response(s) and associated subcase IDs. It is allowed for a DCONSTR card to make reference to a response card that generates multiple responses. Therefore, one DCONSTR card may create hundreds or even thousands of design constraints.

2.2.3 Overall computational efficiency

Based on past experiences, over 90% of the computational resources required for structural synthesis are attributed to finite element analyses and sensitivity analyses, so the CPU time required in the synthesis modules will not be a significant factor for practical problems. Consequently, the measure of overall computational efficiency of structural synthesis is the number of finite element analyses and sensitivity analyses required to obtain a near optimal design. The number of required finite element and sensitivity analyses is governed primarily by two factors: one is the distance between the initial design and the optimal design in the design variable space and the other is the degree of nonlinearity of functions involved. Unfortunately, we don't know where the optimal design is until the work is completed, and the degree of nonlinearity of implicit functions can be predicted only after we start gathering analysis results.

The basic strategy to reduce the number of finite element and sensitivity analyses follows the scheme outlined in reference 3 and summarized in Fig. 2.

We accept that numerical optimization algorithms require relatively large numbers of function evaluations during the numerical search. However, evaluation of functions may not have to be very accurate a long as they are good enough to guide the design to the right direction. Therefore, the results of finite element and sensitivity analyses are used only to generate data to evaluate approximate objective and constraint functions. These approximate functions are <u>explicit</u> functions of design variables, thus can be evaluated extremely fast. As shown in Fig. 2, we never allow the optimizer to communicate directly with finite element analysis, instead it has direct interface only with the approximate model which is a collection of approximate explicit analytic functions. Accuracy of the approximate model is guaranteed in the immediate neighborhood of the particular design where the finite element and sensitivity analyses are carried out. As the numerical search

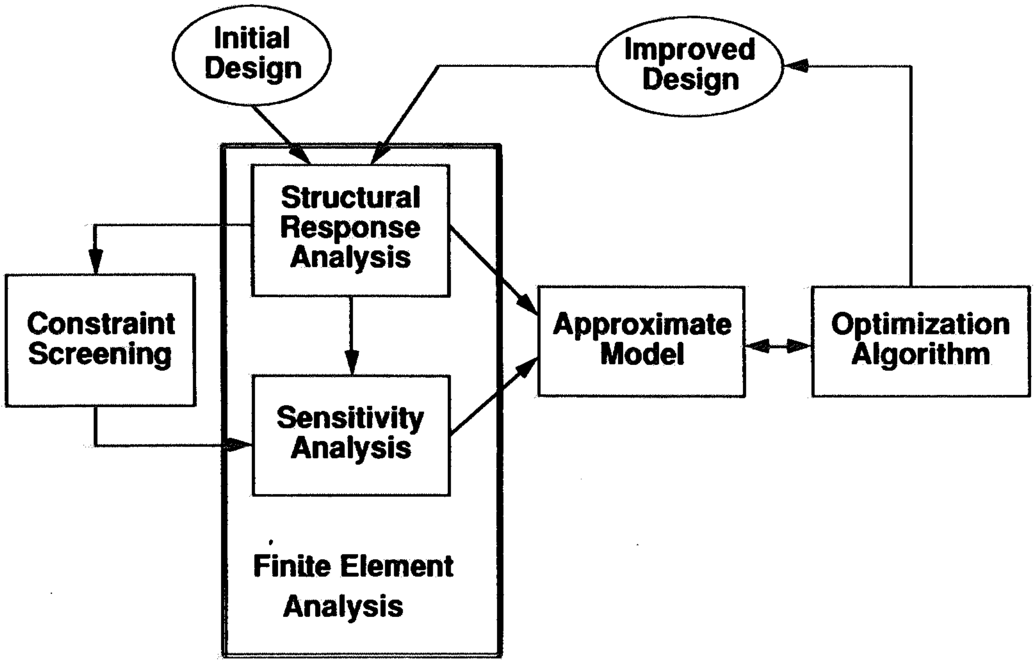

FIG. 2 Strategy to Assemble Efficient Structural Synthesis Program

progresses, the design will start moving away from that design point. The distance we permit the optimizer to move away from the original design depends on our confidence in the quality of our approximate model. In other words, the quality of approximate model is measured by the size of the range in the design space within which accuracy of the approximate model is maintained. Again we don't have sufficient data to quantify what does "accuracy" mean here, but certainly we don't want to find any big surprises when an improved design proposed by the optimizer is analyzed by complete finite element analysis. Accordingly, we like to have approximate functions that maintain "accuracy" in the design variable space for as wide a range as possible.

The simplest form of approximate functions is the first order Taylor series expansion, taking advantage of the sensitivity data. In fact, this is the primary form of approximate model we use in this program. However, straightforward Taylor series approximations do not necessarily provide the best approximation.

Consider an axial stress in a rod element with cross sectional area A carrying axial load P. If the cross sectional area is taken as a design variable, the first order Taylor series approximation is

$$S_{approx} = S_{Ao} + (A - A_o) * \frac{dS}{dA} \qquad (6)$$

Since we know that the axial stress is computed from internal force divided by cross sectional area and that internal force is approximately constant, a better approximation is obtained by using Taylor series approximation with respect to the reciprocal of area, 1/A:

$$S_{approx} = S_{Ao} + \left(\frac{1}{A} - \frac{1}{A_o}\right) * (-A_o^2) * \frac{dS}{dA} \qquad (7)$$

More recently, it was shown [9] that better approximation can be obtained by expanding internal force and then use an exact stress recovery process as:

$$S_{approx} = \frac{1}{A}\left[P_o + (A - A_o) * \frac{dP}{dA}\right] \qquad (8)$$

As shown here, direct Taylor series expansion does not necessarily provide the best approximation and various ideas are considered. The synthesis capability for MSC/NASTRAN will incorporate proven approximate methods as they are developed and verified by numerical simulations. New approximations are more difficult and may require additional data, but are still useful if we can reduce the number of complete finite element analyses even by one. The present capability allows the user three options; direct linearizations, direct and reciprocal approximations based on response type, and direct and reciprocal approximations (mixed variables) based on gradient information. This encompasses each of the current "popular" methods for approximate optimization. Each is a form of linearization, and has been shown to be useful for a broad range of design problems. However, here there is a notable exception to most methods. That is in dealing with type-2 responses, where we have no knowledge of the form of the equation the user will provide. In this case, the type-1 responses that make up the function are calculated according to the chosen approximation method. Then the type-2 response is created from that. Thus, we are able to retain most of the nonlinearity of the user-defined functions.

2.2.4 Open-ended program and data structure

While considering the bulk data format for design model description, we tried not to be constrained by the current capability of DSA modules, or by our present knowledge of structural synthesis. Instead, in principle, design variables are related to any real parameters appearing in the analysis model bulk data. Especially, we expect that an external shape design capability will be implemented in the near future. Thus, the data structure was designed to accept grid coordinate data in the convenient manner for shape design. Similarly, the current set of structural responses is limited, but any responses for which sensitivity data are available should be incorporated in the design process. Therefore, designation of structural responses and description of design constraints are clearly separated. The CONSTR cards for DSA are functionally equivalent to DRESP1 cards for synthesis and DCONSTR cards serve to formulate design constraints.

Internal tables and programs are also designed with future extensions in mind, although a detailed discussion on them is beyond the scope of this paper.

2.3 Design Optimization

As stated previously, the numerical optimization algorithm is applied only to the approximate models, hence it is not too important what type of algorithms are used, as long as the program is well written and has satisfactory robustness. Based on this reasoning, the optimizer installed in the first release will be the modified method of feasible directions [10], instead of the complete set of algorithms available in ADS [11]. The user doesn't have to worry about selection of algorithms and most of the default parameters associated with the modified method of feasible directions don't have to be altered. In consequence, the important decisions to be made by the user are limited to the following:

(1) How far the process should be carried out?
(2) What move limits are used for each iteration?
(3) Selection of threshold values for constraint screening.

2.3.1 Design process control

The user may carry out complete optimization loops till convergence is obtained or may stop at any point to check the results and then to restart. For large problems, it is recommended to intervene, at least, at the end of optimization with respect to approximate models or at the end of finite element analysis, to check the status. For strength design of structures that consist of membranes and bars, a fully stress design loop is also available.

2.3.2 Move limits

For optimization with respect to a design model, analysis variables are not allowed to change by large magnitudes so that errors associated with the approximation don't overwhelm the overall design process. Move limits are imposed as the fractional change allowed to each property, the default value being ± 20%. Selection of move limits will require experience on particular types of problems, thus careful observation on the behavior of structural responses as the design is changing will be important. If the move limit is too small, the number of finite element analyses will grow, and if it is too large, convergence characteristics could be deteriorated or convergence may not be reached.

2.3.3 Constraint screening control

Since the total number of constraints can be extremely large when stress/strain constraints are involved in the design model, it is not desirable nor necessary to carry all of them to sensitivity analysis and then to generation of approximate functions. In order to delete non-critical constraints temporarily, two levels of screening are installed. First, all normalized constraint values are compared against the prescribed threshold values and less critical constraints are deleted. Next, number of retained constraints belonging to the same "group" is limited to the specified level, by deleting less critical constraints within the group. The group is defined differently depending on the type of constraints. For example, stresses associated with elements referencing the same property card and the same subcase are judged to belong to the same group.

Screening threshold values and the maximum number of constraints retained in a group are given in DSCREEN cards. The screening process is repeated on each cycle through the optimization process so that responses that become critical as part of the optimization search process will be retained in later stages.

3. SYNTHESIS PROGRAM ARCHITECTURE

An extremely simplified schematic diagram of the design synthesis system is given in Fig. 3. The names given in each block of this diagram do not represent actual modules, in fact most of them consist of multiple number of modules.

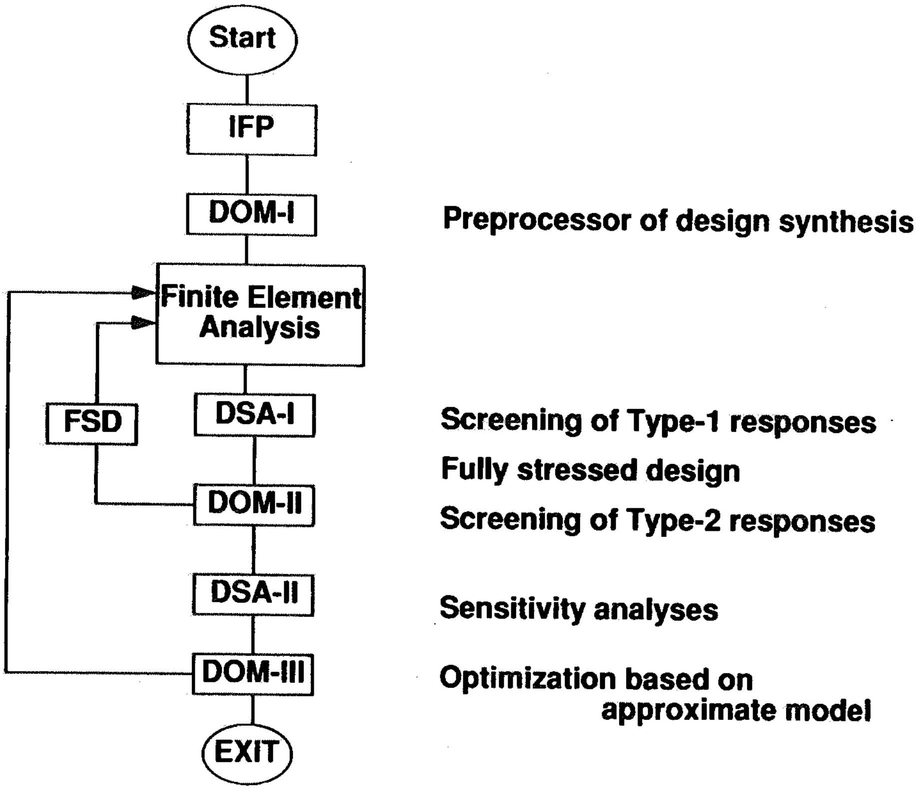

Fig. 3 Simplified Schematic Diagram of Program Flow

The DOM-I block initializes tables and matrices based on the user supplied data. All data that are not affected by the changes of values of design variables are prepared in this block. The DSA-I retrieves all the Type-1 responses required in the design model and performs initial screening against the threshold levels. DOM-II performs screening on Type-2 responses and identifies Type-1 responses that are associated with retained Type-2 responses. DSA-II carries out sensitivity analysis on retained Type-1 responses with respect to independent design variables as well as analysis model parameters described by DVPREL2 and DVPREL3 cards. The functions of DOM-III are to set up the approximate models, carry out optimization with respect to approximate models, make judgments on convergence status, to print out results and, if requested, update the analysis model parameters in the database to reflect the modified design variables.

The user is requested to prepare three types of decks as a standard MSC/NASTRAN data deck, i.e. Executive control, Case control and Bulk data decks. For executive control, a new rigid format is prepared to perform all required structural analyses and sensitivity analyses within one rigid format. Preparation of case control and bulk data decks may be considered identical to conventional MSC/NASTRAN decks, except that bulk data deck must now include design model data in addition to analysis model data.

4. CONCLUDING REMARKS

While finite element structural analysis has been used successfully to predict responses of given structures, the synthesis capability now being installed adds a new dimension to control structural responses by providing assistance in the actual design process.

The basic philosophy and methodology used in automated synthesis are easy to understand intuitively, because they may be regarded, in our opinion, as nothing but comprehensive mechanization of what excellent design engineers have been doing all the time. However, by taking advantage of capabilities of modern computers, this technology is impacting computer aided engineering to such an extent that effective use of optimization may become a critical factor in a competitive market. Also innovative use of the synthesis capability provides high level information such as identifying where research investment should be made to yield the best return. We expect the new synthesis capability will stimulate creativity of the users and give us critical reviews as well as positive feedback to achieve the ultimate objective - the best engineering design.

5. REFERENCES

1. Vanderplaats, G.N. "CONMIN: A FORTRAN Program for Constrained Function Minimization", NASA TM-X 62,282 Aug. 1973
2. Vanderplaats, G.N. "ADS: A FORTRAN Program for Automated Design Synthesis" Engineering Design Optimization, Inc., Santa Barbara, Calif. May 1985
3. Schmit, L.A. and Miura, H. "Approximation Concepts for Efficient Structural Synthesis", NASA CR-2552, March 1976
4. Narayanaswami, R. "Design Improvements Using Design Sensitivity Analysis Capability of MSC/NASTRAN", Proceedings of MSC/NASTRAN Users' Conference Pasadena, Calif. March 1984

5. "ADS/NASOPT User's Manual", CSA Engineering, Palo Alto, Calif. and Engineering Design Optimization, Santa Barbara, Calif. 1986
6. Chargin, M.K., Miura, H. and Clifford, G. "Dynamic Response Optimization Using MSC/NASTRAN", Proceedings of MSC/NASTRAN World Users' Conference, Los Angeles, Calif. March 1987
7. Botkin, M.E., Lust, R.V., Yang, R-J., Song, J.O. and Katnik, R.B., "Structural sizing Optimization Using an External Finite Element Program", AIAA-87-0883-CP AIAA/ASME/ASCE/AHS 28th SDM Conference, Monterey, Calif. April 1987
8. Inoda, K., Sugihara, T., Murakami, K. and Kakuya, T. "Development of Structural Identification and Optimization System", Proceedings of 5th MSC/NASTRAN Users' Conference in Japan, Tokyo, Japan, Oct. 1987
9. Vanderplaats, G.N. and Salajeghah, E. "A New Approximation Methods for Stress Constraints in Structural Synthesis", AIAA/ASME/ASCE/AHS 28th SDM Conference, Monterey, Calif. April, 1987
10. Hansen, S. R., Micro-Dot User's Manual, Engineering Design Optimization, Inc., Santa Barbara, CA, 1987
11. Vanderplaats, G. N., and Sugimoto, H., "A General-Purpose Optimization Program for Engineering Design," International Journal of Computers and Structures, Vol. 24, No. 1, 1986.