REPORT NO.
UCB/SESM-84/8

STRUCTURAL ENGINEERING AND
STRUCTURAL MECHANICS

STRUCTURAL OPTIMIZATION OF DISCRETE SYSTEMS REPRESENTED BY FINITE ELEMENTS

by

DIETRICH HARTMANN

JUNE 1984

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

Structural Optimization of Discrete Systems Represented by Finite Element Models

Dietrich Hartmann *

Department of Civil Engineering
Structural Mechanics and Structural Optimization
University of Dortmund, Dortmund, West-Germany

ABSTRACT

The purpose of this report is to demonstrate how the coupling of a finite element model and a nonlinear optimization model leads to a new compound model, the structural optimization model. In addition, this report is the conceptual and logical base for the creation of a general purpose structural optimization and repetitive analysis program, which has been partially developed parallel to this report.

It is shown that the applied optimization method has to be generally applicable. Otherwise, a large number of modifications would be necessary to adjust the finite element method to a special optimization problem and vice versa. However, such modifications can be extremeley expensive and would be hardly justifiable. Therefore, narrowly based optimization techniques are not appropriate for the practical optimization of structural systems represented by finite element models.

It is demonstrated that, at present, the family of evolution strategies seems to be the best optimization technique available. These strategies represent a reasonable compromise between global convergence, convergence rates, reliability and numerical effort. Since optimization means that the optimal solution has to be found iteratively the structural analysis has to be repeatedly performed. This re-analysis process regularly causes an extraordinary increase of computational effort. Subsequently, the re-analysis formulation of the finite element equations forms an important interface between optimization and finite element model.

Within the scope of this contribution, only discrete linear elastic structural systems are considered. The structural optimization model worked out, however, encompasses geometrical as well as structural optimization variables while most of the contributions in this field only deal with structural variables.

July 4, 1984

* Professor Dr.-Ing.; at present, visiting scholar in Structural Engineering and Structural Mechanics, Department of Civil Engineering, University of California, Berkeley, U.S.A., with support from the Alexander-von-Humboldt-Foundation.

ACKNOWLEDGEMENT

This report has been prepared during a stay as visiting scholar in Structural Engineering and Structural Mechanics at the University of California, Berkeley.

The author wishes to express his sincere appreciation for the generous support and fellowship awarded by the Alexander-von-Humboldt-Foundation, which enabled him to perform a research project entitled " Structural Optimization of Finite Element Models ". Portions of this research project form a constituent part of the present report.

Thanks are also due to the Department of Civil Engineering, University of California, Berkeley, for supporting the research project by providing computer facilities.

The author wishes to gratefully acknowledge Prof. G.H. Powell, Department of Civil Engineering, University of California, Berkeley, for his co-operation.

Dietrich Hartmann

Berkeley, U.S.A., June 1984

TABLE OF CONTENTS

1	Introduction	1
2	Design problem	1
3	General comments on optimization models	2
4	Fundamentals of the formulation of optimization problems	3
5	Formulation of the structural optimization model	5
6	Selection criteria for optimization techniques	6
7	Description of the applied optimization technique (evolution strategies)	11
7.1	General comments	11
7.2	Convergence behaviour	11
7.3	Theoretical concept of the evolution strategies	12
7.3.1	Generation process	13
7.3.1.1	(1+1)-strategy	13
7.3.1.2	Multimembered $(\mu+\lambda)$ and (μ,λ)-strategies	17
7.3.2	Selection process	18
7.3.2.1	(1+1)-strategy	18
7.3.2.2	(μ,λ)-strategy	19
7.3.3	Adjustment of the optimization process	20
7.3.3.1	(1+1)-strategy	20
7.3.3.2	(μ,λ)-strategy	22
7.3.4	Convergence and termination criteria	24
7.3.4.1	(1+1)-strategy	24
7.3.4.2	(μ,λ)-strategy	25
8	Re-analysis formulation of the finite element model	26
8.1	Necessity	26
8.2	Numerical effort of the total structural optimization model	26
8.3	Approximation schemes at overall structural system level	28
8.3.1	Linear finite element equations	29
8.3.2	Eigenvalue problems	34
8.4	Relationships at element and optimization-variable level	35
9	Conclusions	38
10	Overlook on future expansions and improvements	38
11	References	39

1. Introduction

Both the method of numerical optimization of structures and the finite element approach were introduced at exactly the same time. At the second ASCE conference on Electronic Computation, held at Pittsburg 1960, CLOUGH [1] presented a paper on the finite element approach using the term finite element for the first time, and SCHMITT [2] gave a lecture on structural optimization. Both contributions have had a lasting influence on the field of engineering research.

Undoubtedly, the finite element method has emerged as a powerful method in structural analysis. One of its most important advantages is its general applicability for different types of structures.

Structural optimization, however, still has to struggle for recognition. To the author's viewpoint, this has to do with the lack of a general underlying concept. As a consequence of this deficiency, innumerable different solution methods have been developed for a large number of special optimization problems. Although individual solution techniques may be justified from the viewpoint of numerical efficiency in special cases, a more general basis is urgently needed to facilitate the coüpling of the finite element method and numerical optimization towards a structural optimization model.

By linking both models every design problem can be formulated as an optimization problem, which then can be numerically solved. Consequently, the numerical optimization of structural systems associated with an appropriate finite element model can be regarded as a natural base for solving design problems in a computer-oriented manner. Structural optimization, therefore, provides a powerful computer-aided design tool which facilitates and enhances the decision making process in the field of engineering.

2. Design problem

In order to explain the difference between a conventional and a structural-optimization-aided design procedure let us consider an example. To design a multi-storey building usually involves finding a structural solution that fits all requirements and demands given, such as economical, technical and constructional ones. Therefore, many different aspects must be considered to find an appropriate solution for the current problem.

In the conventional design process an initial trial of the structural design is made. To predict the performance of the structure during and after construction a computational model is choosen. A suitable model is the finite element model. This model provides some insight into the structural behaviour. Having obtained a solution the results must be checked. If the solution is not adequate the input data for the model must be modified. If large scale structures are considered the expense to re-prepare the input data due to changes desired can be considerable. Additional effort has to be made to control re-runs of the finite element program. Moreover, the output results have to be checked and interpreted after each modification. Therefore, pre- and post-processors have recently been developed to support the engineer's work. Although both types of processors provide valuable help, they are, however, not capable of automatizing all the essential parts of the design process. The control of repeated standard data-preparation and finite element re-runs, as well as the standard verfication of results is still incumbent on the engineer, even if pre- and postprecessors are available.

On the contrary, using a structural optimization model makes possible the computerization of most parts of the design process. The complete re-analysis process and evaluation of the output results of the finite element program, as well as the evaluation of the structural quality can be automatically accomplished. In effect, the structural optimization model itself acts not only as a processor but does so in a more comprehensive manner than the already known pre- and postprocessors do.

The advantage of an automatized design procedure is quite obvious. The engineer is provided with optimal solutions for structures without the necessity of repeating standard input-

data-preparation and evaluation of output results. Thus, structural optimization saves valuable manpower and enables the engineer to design more improved structures more rapidly.

3. General comments on optimization models

The structural optimization model essentially depends on four fundamental features,

(1) the formulation of the optimization model in terms of the optimization variables (or design variables), the objective function or optimization criterion and the constraints;

(2) the numerical behaviour of the applied optimization strategy and its range of applicability;

(3) the compatibility between the finite element model and the numerical optimization model;

(4) the re-analysis process formulated by means of finite element equations and controlled thr h the optimization routine.

These four points are discussed in more detail in the subsequent chapters. To improve the understanding of the reader, the general philosophy taken as a basis is pointed out in advance. Therefore, some general comments are made to elucidate the formal concept of the choosen philosophy.

As previously mentioned, the structural optimization model consists of two sub-models. The first is the optimization model, the second is the finite element model. However, both models are very closely related to another. Hence, the formulation of the optimization model directly affects the finite element model and vice versa. Recently, several papers were published in which both models were coupled. In most cases, "only" two different software packages were linked together. What is needed, however , is a more sophisticated approach based on theoretical contemplations. For this reason, in the present report an attempt is made to elaborate a logical and natural base for the coupling of both models.

In this context, it should be pointed out that the term coupling precisely means the enbedding of the finite element model into the numerical optimization model. Hence, the structural analysis process represented by a finite element model only forms a small, however, an important part of the total structural optimization model. Consequently, the structural optimization model must be regarded as a model of higher rank, being more comprehensive and of "superior" significance than the finite element model. Therefore, it makes sense that the optimization model must be at least as generally applicable as the finite element model.

Correspondingly, it seems unreasonable to enbed the finite element model into a narrowly based optimization model which, for example, is perhaps only applicable to a linear or quadratic optimization problem. Then, the effort required to make the finite element model an integral and compatible part of the total structural optimization model would not be justified. Since technical optimization problems are regularly non-linear and non-quadratic all the special modifications necessarily incorporated to save storage and computer-time would be futile if the optimization problem is of higher order than linear or quadratic. Unfortunately, practical problems do not take care of defined mathematical categories of nonlinear problems and often result in intricate situations, especially, if geometrical optimization variables are considered. In that case, objective functions with multiple "peaks" and disconnected feasible domains, which perhaps contain local optima, can frequently occur. A typical and representative example is worked out by HOEFLER [3] . He could show that, already in the simple case of geometrical optimization of statically determined trusses, the objective function has infinite barriers dividing the feasible region into several individual subdomains with potential local optima. Further problems which are in fact more complicated have been worked out by the present author [4], [5], [6], [7]. Accordingly, it is postulated that the optimization model must be capable of covering a wide range of potential problems of different degree of complicacy.

The optimization model has to control and to evaluate the results obtained from the finite element model with respect to the optimization criterion and the current constraints. In addition, it has to determine if re-analysis is necessary or not. Obviously, the re-analysis process represents an important interface between both models because of its feedback character and its

effects on the computational effort. Actually, even medium-sized design problems require many re-analysis cycles. Hence, it is crucial that the re-analysis formulation represented by finite element equations use computer resources economically. As we know, the analysis of structural systems with a major degree of freedom can already be very time and memory-consuming if a conventional finite element analysis is performed. Numerical optimization algorithms, however, demand the values of the optimization criterion and constraints, conditionally, hundreds of times before an optimal solution is found. Since each evaluation of the contraints or the optimization criterion may be associated with a complete or incomplete re-analysis process, the dilemma encountered is obvious. If we couple both the finite element and the optimization model, conditionally, an extraordinary time and memory-consuming problem can arise due to the multiple re-analysis problem.

According to the comments outlined above, the conclusion can be drawn that two key issues are substantial for more advanced optimization models

(a) general applicable optimization methods,

(b) suitable re-analysis formulations.

4. Fundamentals of the formulation of optimization problems

If we want to establish structural optimization models which are distinguished by general applicability, we have to formulate the optimization problem differently from conventional formulations.

Usually, the nonlinear optimization problem is defined - without restricting generality - as a minimum problem using the following formulation :

$$f(x_1,x_2,\ldots,x_n) \rightarrow Minimum \tag{4.1}$$

subject to the constraints

$$\begin{aligned} &g_j(x_1,x_2,\ldots,x_n) \geqslant 0 \\ &j = 1,2,\ldots,m \end{aligned} \tag{4.2}$$

where the real-valued functions f and g_j are the objective function and the constraints, respectively. Both types of functions depend on the optimization or design variables $x_1,x_2,\ldots,x_n$. These variables are formally concentrated in the design vector $\mathbf{x}$ representing points in the n - dimensional Euclidian space E^n, where

$$\mathbf{x} = \left\{ x_1,x_2,\ldots,x_n \right\}^T \tag{4.3}$$

If we also consider problems which simultaneously contain inequality constraints $g_j(\mathbf{x}) \geqslant 0$ and equality constraints $h_k(\mathbf{x}) = 0$, $k = 1,2,\ldots,l$, we have

$$f(\mathbf{x}) \rightarrow Minimum \tag{4.4}$$

subject to the augmented set of constraints

$$\begin{aligned} &g_j(\mathbf{x}) \geqslant 0 \quad \text{and} \quad h_k(\mathbf{x}) = 0 \\ &j = 1,2,\ldots,m \\ &k = 1,2,\ldots,l \end{aligned} \tag{4.5}$$

If we use a more compact minimization formulation and write the functions $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ in matrix form we obtain

$$\min_{\mathbf{x}\in S^n} \left\{ f(\mathbf{x}) \left[\begin{array}{ll} \mathbf{g}(\mathbf{x}) & \geqslant 0 \\ \mathbf{h}(\mathbf{x}) & = 0 \end{array} \right. \right\} \tag{4.6}$$

The above relationship indicates that the minimum of the real-valued function $f(\mathbf{x})$ is sought within a feasible domain S^n, which is a subset or subspace of the Euclidian space E^n and defined by means of the constraints $\mathbf{g}(\mathbf{x})$. Additionally, the vector $\mathbf{x}$ in the feasible domain must satisfy the equations $\mathbf{h}(\mathbf{x})$.

Within conventional nonlinear optimization, which is mainly based on mathematical-oriented techniques, it is demanded that the vector $\mathbf{x}$, the function $f(\mathbf{x})$ and the constraints $g(\mathbf{x})$ must fulfill specified requirements in order to be able to apply to special types of numerical optimization routines. To meet such requirements assumptions are established. Some of these assumptions frequently made are for example:

(a) the function $f(\mathbf{x})$ has to be linear or convex, and/or differentiable;

(b) the constraints $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ also have to be linear or convex, and differentiable for each $\mathbf{x}\in S^n$;

(c) the optimization problem must have, by definition, one and only one minimum within the feasible domain S^n;

(d) the feasible domain S^n must be a simple connected domain with sufficiently regular boundary.

In particular, the assumption of differentiability is made because a large number of numerical solution techniques need first, and conditionally second derivatives or approximations of the derivatives of the functions $f(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$. But, general engineering optimization often involves problems with nondifferentiable quantities, and also convexity or linearity is confined to special cases. Thus, in most cases practical problems lead to noncategorizable relationships for the objective function $f(\mathbf{x})$ and some of the constraints $\mathbf{g}(\mathbf{x})$. As stated above, the simple optimization of statically determined trusses with variable geometry already yields a tremendously intricated problem. Therefore, many of the conventional mathematical assumptions usually made are not true of real world problems.

Consequently, the assumptions of differentiability, convexity and unimodality (existence of only one optimum) should be dropped because these assumptions prevent the development of advanced structural optimization models. For that reason, a more general definition of the problem (4.6) will be employed

$$\min_{\mathbf{x}\in S^n} \left\{ alg{\rightarrow}f(\mathbf{x}) \left[\begin{array}{ll} alg{\rightarrow}\mathbf{g}(\mathbf{x}) & \geqslant 0 \\ \mathbf{h}(\mathbf{x}) & = 0 \end{array} \right. \right\} \tag{4.7}$$

where $alg{\rightarrow}f(\mathbf{x})$ and $alg{\rightarrow}g(\mathbf{x})$ indicate that algorithmical formulations of $f(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are allowed. The problem (4.7) is called an algorthmically nonlinear problem. The term "*alg*" has the meaning of an operator and demonstrates that the functions $f(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ must not necessarily belong to the C^1 or C^2 class. (This means that they must not necessarily be differentiable). The symbol "→" denotes that the operator "*alg*" represented by an appropriate algorithm acts on the term "$f(\mathbf{x})$" following the symbol "→".

There are also some further numerical aspects which imply that methods based on derivatives ought not to be applied. As shown by CARPENTER and SMITH [8] a large numerical effort is necessary to obtain derivatives of the objective function or augmented objected function. First derivates are computationally expensive and second ones even more so. Furthermore, in most cases derivatives can not directly be calculated by hand since that would be too cumbersome. Therefore, derivatives have to be found from finite difference formulae. However, it can be observed that this procedure easily leads to ill conditioning and divergence due to truncation and cancellation errors (see LIPSON and GWIN [9]).

A further crucial disadvantage is the fact that all well known numerical optimization methods based on derivatives have to face difficulties if multiple local optima occur. The reason for this is that all these methods make use of the a-priori-assumption that one and only one optimum exists. However, this assumption is not true in a lot of practical cases.

5. Formulation of the structural optimization model

Now, let us extend the "pure" optimization model (4.7) to a structural model by incorporating the finite element model. This can be accomplished by replacing the formal equality constraints $\mathbf{h}(\mathbf{x})$ by the governing finite element matrix equations. Of course, these equations depend on the type of structure and its structural behaviour. In this contribution only discrete systems with linear behaviour are taken into consideration. However, an analagous assignment to more complicated structures is possible using the same logic.

In the case of discrete systems with linear structural behaviour we have

$$\min_{\mathbf{x}\in S^n} \left\{ alg{\rightarrow} f(\mathbf{x}) \left[\begin{array}{l} alg{\rightarrow} \mathbf{g}(\mathbf{x}) \geqslant 0 \\ \mathbf{K}(\mathbf{x})\ \mathbf{u}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \\ \mathbf{K}(\mathbf{x})\ \mathbf{v}(\mathbf{x}) = \lambda(\mathbf{x})\ \mathbf{M}(\mathbf{x})\ \mathbf{v}(\mathbf{x}) \end{array} \right. \right\} \tag{5.1}$$

As it can be seen, in the relationship (4.7) the equation system $\mathbf{h}(\mathbf{x})=0$ is replaced by two distinct equation systems. The first one represents the governing linear finite element equations of the form

$$\mathbf{K}(\mathbf{x})\ \mathbf{u}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \tag{5.2}$$

where

$\mathbf{K}(\mathbf{x})$ = global stiffness matrix of the structural system depending on the optimization vector $\mathbf{x}$

$\mathbf{u}(\mathbf{x})$ = structural response expressed in terms of the generalized displacements, also depending on the vector $\mathbf{x}$

$\mathbf{P}(\mathbf{x})$ = generalized load vector depending on the vector $\mathbf{x}$ if the load can change with respect to variations of the vector $\mathbf{x}$.

The second one represents an eigenvalue problem which may be involved if critical values (e.g. critical loads) have to be calculated within the optimization process. The corresponding eigenvalue problem can be described in its generalized form such as

$$\mathbf{K}(\mathbf{x})\ \mathbf{v}(\mathbf{x}) = \lambda(\mathbf{x})\ \mathbf{M}(\mathbf{x})\ \mathbf{v}(\mathbf{x}) \tag{5.3}$$

where

$\lambda(\mathbf{x})$ = eigenvalues depending on the optimization vector $\mathbf{x}$

$\mathbf{v}(\mathbf{x})$ = eigenvectors representing the appropriate structural response and depending on the vector $\mathbf{x}$

$\mathbf{M}(\mathbf{x})$ = real valued matrix depending also on the vector $\mathbf{x}$.

For example, in vibration analysis the matrices $\mathbf{K}$ and $\mathbf{M}$ are stiffness and mass matrices whereas the eigenvalues represent the square of frequencies; if critical buckling loads are to be calculated, the matrices $\mathbf{K}$ and $\mathbf{M}$ can be thought of as first order terms of the (mechanically) nonlinear stiffness matrix whereas λ are the critical values of the buckling load. Mathematically, solving the eigenvalue problem is equivalent to calculating the roots of the polynomial $p(\lambda)$, which has order equal to that one of the matrices $\mathbf{K}$ and $\mathbf{M}$.

Obviously, the equation systems (5.2) and (5.3) represent a re-analysis problem because the responses **u** and **v** depend on the vector of the optimization variables (design vector) **x**, which must be continuously updated, according to the logic of the applied optimization algorithm. On the other hand, a major number of the constraints $alg \rightarrow \mathbf{g}(\mathbf{x})$, of course, depend on the structural response **u**, **v** and λ (e.g. stress, displacement , stability or frequency constraints). To be more thorough, therefore, the relationship (5.1) must be re-written, and then we obtain the structural optimization model in its general form

$$\min_{\mathbf{x} \in S^n} \left\{ alg \rightarrow f(\mathbf{x}) \; \middle| \; \begin{array}{l} alg \rightarrow \mathbf{g}(\mathbf{x},\mathbf{u},\mathbf{v},\lambda) \geqslant 0 \\ \mathbf{K}(\mathbf{x})\,\mathbf{u}(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \\ \mathbf{K}(\mathbf{x})\,\mathbf{v}(\mathbf{x}) = \lambda(\mathbf{x})\,\mathbf{M}(\mathbf{x})\,\mathbf{v}(\mathbf{x}) \end{array} \right\} \tag{5.4}$$

The advantage of the applied formulation (5.4) is that the total optimization model is displayed within a single relationship , and the interdependencies of the associated sub-models are indicated at first glance.

It should be emphasized that the scalar λ, the vectors **u** and **v** as well as the matrices **K** and **M** are implicit functions of the design vector **x** which accounts for a design change (change in geometry or the structural data). In order to determine whether a current solution **x** is feasible or not all structural equations must be previously solved, completely or approximatively.

The re-analysis problem represented by Eqs. (5.2) and (5.3) forms the already mentioned interface between both the numerical optimization model { see relationship (4.6) , (4.7) } and the finite element model in its standard form

$$\mathbf{K}\,\mathbf{u} = \mathbf{P} \tag{5.5}$$

and

$$\mathbf{K}\,\mathbf{v} = \lambda\,\mathbf{M}\,\mathbf{v} \tag{5.6}$$

where all the elements of the corresponding matrices are constant real values and not functions depending on **x** vectors.

In conventional nonlinear programming it is assumed that the constraints depend explictly on the optimization variables, and that this dependency is known. Unfortunately, this is not the case with structural optimization problems because of the implicit nature of the finite element equations representing the re-analysis process. As a consequence, also the structural constraints (e.g. constraints for stresses, displacements, forces, etc.) are implicit relationships. The implicit form of these constraints is rather a vexing problem and forms a severe obstacle. The time-consuming calculation of implicit constraints can only be reduced if the number or the computational effort of the re-analysis processes is reduced. Reduction schemes approximating the corresponding finite element equations (5.2) and/or (5.3) , respectively, are discussed in chapter 8.

First, however, we will be discussing criteria to select an appropriate optimization strategy capable of solving problems of the general form (5.4). Then in chapter 7, based on these criteria, the evolution strategies will be introduced which are a powerful optimization method for structural optimization.

6. Selection criteria for optimization techniques

The computational efficiency of an optimization method, to some extent, depends on the personal point of view. Consequently, there are always several competitive methods, and it is

questionable if there will ever be solely one strategy capable of solving all potential problems with the same maximum degree of efficiency. So far it has been advisable to employ a special category of solution techniques if a given problem falls into such a category, and must be solved most efficiently. However, general structural design problems differ considerably from conventional optimization problems because we cannot categorize the majority of them. As repeatedly emphasized, in this field general applicability is one of the most important criteria, among others, for two main reasons which are once more summarized

(1) real world structures lead to optimization problems for which conventional idealizations , such as convexity, differentiability, unimodality, are not valid in most cases.

(2) modifications of the finite element model, and vice versa, measures to adapt the optimization model to the finite element formulation are only worthwhile if the optimization model covers a wide range of problems.

If we consider the feature "general applicability" as crucial we can omit from our selection all methods using derivatives, and focus on derivative-free methods. To decide which of these methods is the best, further criteria have to be considered. The dilemma, however, is that it is difficult to compare all potential methods because a regrettable lack of uniformity in the presentation of numerical optimization results exists. To give some examples, in most cases the numerical accuracy required is not published; in some cases the execution time is omitted or the used computer is not indicated.

An outstanding evaluation of competitive derivative-free methods was recently published by SCHWEFEL [10]. To the author's knowledge this is the only publication which forms a reasonable basis of comparison because all the usual imponderabilities are excluded. All compared algorithms (14 popular methods) were programmed using the same logic ,and the following numerically significant criteria were evaluated:

-computational effort

-storage demand

-convergence rate

-global convergence with respect to multiple optima

-numerical reliability

In this context, it should be outlined why the quantity numerical effort is not as crucial as usually assumed. If the simpliest case of nonlinearity, a simple quadratic problem without constraints, is considered it can be shown that the order of the numerical effort of all popular optimization techniques cannot be less than $ italic O (n sup 3) $ where $ n $ denotes the number of the optimization variables. The following table (Fig. 6-1) illustrates the individual coherences in more detail, comparing the three main categories of optimization strategies (Newton methods, gradient methods and derivative-free methods).

Opt. method	no. of iterations	operations per iteration step			
		calc. of obj. funct.	calc. of gradients	calc. of Hesse matrix	element. operat.
Newton	n^{0*}	0	n^0	n^0	n^3
Gradient	n^1	n^0	n^0	0	n^1 or n^2 resp.
Deriv. Free	n^2	n^0	0	0	n^1
weighting factors		n^0	n^1	n^2	./.

* ($n^0 = 1$)

Fig. 6-1: Computational effort

In table Fig. 6-1, first Newton methods are considered. As it can be seen the optimum solution is found within one iteration step (n^0). On the other hand, the evaluation of the gradient vector $\frac{\partial f}{\partial \mathbf{x}}$ and the Hesse matrix $\frac{\partial^2 f}{\partial \mathbf{x}^2}$ is necessary. That means that ½ n $(n+1)$ second partial derivatives are needed. The most expensive numerical operation, however , is the inversion of the Hesse matrix which demands a computational effort of the order O (n^3). Secondly, gradient methods need n linear one-dimensional searches. For each search the calculation of the gradient method is to be performed, additionally, the objective function is to be calculated per each iteration step. The number of elementary arithmetic operations is of the order O (n) or O (n^2), depending on the applied alternative, because of the multiplication of the corresponding matrices and vectors. Thirdly, derivative free methods are considered. These methods need a greater number of iteration steps because they do not utilize information based on derivatives to determine search directions. However, the number of elementary operations per iteration step is only of the order of O (n), and no gradients have to be calculated.

To compare the individual categories of the methods mentioned above, we need a measure of comparison. Following a proposal made by FLETCHER [11] we equate the calculation of a Hesse matrix to n calculations of a gradient, and to n^2 calculations of the objective function. Thus

$$F : \frac{\partial f}{\partial \mathbf{x}} : \frac{\partial^2 f}{\partial \mathbf{x}^2} \sim n^0 : n^1 : n^2 \tag{6.1}$$

Therefore, the total effort for solving a given optimization problem is at least of the order O (n^{2+FAK}) where the term FAK represents the numerical effort to evaluate the current objective function. (Newton : $n^2+n^0 \approx n^2$, Gradient : $n^1+n^1=n^2$, Derivative Free: $n^0+n^2 \approx n^2$). In the assumed simple quadratically nonlinear case, we can estimate the factor FAK to be equal to "1"; (e.g., if a more general quadratic function of the form $\mathbf{x}^T \mathbf{A} \mathbf{x}$ with a fully populated matrix $\mathbf{A}$ occurs the term FAK would be exactly equal to "2"). Hence, we obtain for the minimum computational effort

$$T \sim n^{2+FAK} \geqslant n^{2+1} = n^3 \qquad (6.2)$$

where the term T denotes the execution time of the computer program; the symbol "~" labels that the time T is proportional to the order of the subsequent arithmetical expression. In other words, the computational effort of all known optimization methods increases at least as the cube of the number n of the optimization variables.

If the computational effort for all known methods, however, is approximately of the same order this feature is not sufficient , and the selection of the best method must be based on other reasons. A more realistic quantity is the product of numerical effort and main-storage-demand (main-storage-demand due to the length of the optimization modul and the used storage locations). This quantity (measured in Kilo-Bytes times sec = KBsec) is of particular significance if large scale structures are to be optimized. In such cases, often the finite element analysis already reaches the physical limits of the computer with the consequence that special analysis techniques (out-of-core equation solver, databased management routines, partitioning, substructuring, etc.) must be used. If the applied optimization method also demands a very large number of storage locations, these limits are reached even earlier , and some of the special methods mentioned above are needed still earlier. That , however, would degrade the efficiency of the total structural optimization model because of the low speed of out-of-core operations performed.

Reliability of an optimization method is also a very important feature. This quantity depends on robustness in operation , numerical stability and error propagation associated with the solution technique. The more a method is based on elementary numerical operations, the more robust and reliable it is. Reliability is especially important if a large number of optimization variables is involved, and the topology of the feasible optimization space is complicated.

The most significant criterion, however, is the ease with which a technique can be modified and adjusted to changing needs, such as changes in the nature of problems (multi-criterion optimization) and computers (multi-processor systems). The most popular optimization methods presently applied are sequentially organized methods. However, recently, computers with parallel processors, sharing the same main memory, have been developed. These computers can perform several distinct tasks or calculation steps parallelly and simultaneously. Therefore, they can run programs based on parallelly organized solution techniques that will have further impacts on future optimization and finite element calculations. Parallel codes can tremendously reduce the total run-time and will lead to more advanced numerical methods. Therefore, it should be checked if a selected method provides vectorizable program-codes that will fit the parallel concept of modern computers.

The ability to adjust to changing needs has to do with artificial intelligence. Using the definitions of artificial intelligence, also allows us to evaluate the quality of an optimization method, which - as already stated - acts as an intelligent processor. The degree of intelligence of such a processor can be judged with respect to its ability to adjust to new problems and technology , and to extract and store information which can be referred to on future occasions to improve its ability to adjust. Six principal built-in-mechanismi can be identified to make a versatile and highly intelligent processor. These mechanismi described in table Fig. 6-2 should be also used for selecting a suitable optimization method.

Built-in-mechanismi	Definition
Data capture ability	ability to extract information from a problem
Data storage ability	ability to store information which can be referred to on future occasions
Processing speed	speed with which the input information can be processed
Software flexibility	ability to modify the software due to changing needs
Software efficiency	logic in which software is structured and organized
Software range	ability to cover a wide range of problems

Fig. 6-2 : Definitions of artificial intelligence

With regard to the above considerations the selection of an appropriate optimization technique was carried out. The result obtained was that the family of evolution strategies, described in the next chapter, represents the best compromise between general applicability, global convergence, reliability and convergence rate for complicated problems. This result could be verified by the present author and several others [3], [4], [5], [6], [12], [13], [14], [15] for a variety of different engineering optimization problems.

From the above discussion it should be clear that structural optimization is an application-oriented subject. Therefore, the selected optimization strategy should also aim at an interactive dialog between the optimization algorithm and the engineer. Since a good engineer is an intuitive optimizer, as we can immediately see from real world problems solved in the last decades, his knowledge and experience must be integrated into an optimization process. However, to avoid confusion it should be emphasized that there is no alternative than numerical optimization if the optimization model encompasses more than, say, three or four optimization variables, and if the experience of the engineer with a particular problem is relatively small.

7. Description of the applied optimization technique (evolution strategies)

7.1. General comments

The family of evolution strategies allows the solution of general restrained optimization problems with highly nonlinear objective criteria and constraints. Both the objective criterion and the constraints may even have an arbitrarily algorithmical nature, and both may depend on a large number of optimization variables. Therefore, the family of evolution strategies can be regarded as a general purpose solution method within the field of engineering optimization. The individual alternative algorithms of this family of methods are all based on the same logical concept. According to the complicacy of the present optimization problem, the single strategies can be applied individually and/or in combination with each other.

Each of the individual strategies represent a direct solution technique through which potential optima can be iteratively found by means of a sophisticated random process. The constraints permitted must be formulated as inequality constraints where, without restricting generality, feasibility is assumed if the right hand side of the constraints is greater than or equal to 0. However, if equality constraints have to be regarded because they can not eliminated by virtue of separate calculations, such equality constraints can be transferred into inequality constraints by using so called slip - variables or Langrangian or penalty formulations.

In the case of structural optimization, we assume that the only equation constraints incorporated are represented by finite element matrix equations , and that these equations can be solved separately from the optimization process. This assumption is advantageous because matured finite element codes are already available which can be efficiently applied to the solution of structural equation systems. Also, we can take advantage of the existing input and output data - bases of the finite element program .

7.2. Convergence behaviour

Convergence of an iterative process ensures sufficiently accurate and rapid results and, therefore, is a very substantial feature. As we know, in the field of structural optimization exceptionally complicated problems may occur. The feasible domain may be a highly dimensional space with an intricate topology and multiple local optima. Hence, nonlinear optimization problems are different from other nonlinear problems, for example, nonlinear problems in the structural analysis field where the nonlinear nature of the problem can be more accurately anticipated. Particularly, the quantity "convergence rate", which represents the speed of iteration and is usually considered as the most essential quantity, only characterizes one aspect among further ones if we have to decide on the quality of optimization strategies.

The philosphy of the evolution strategies is to provide numerical methods which are as generally applicable and flexible as possible , and which have reasonable convergence behaviour. However, the features "reliability", "robustness in operation" and "global convergence" are higher ranked than "convergence rate". With regard to convergence rate, the evolution strategies may possibly yield poor convergence rates in special problems,particularly, if linear or quadratic problems have to be solved. It must also be outlined , however, that in the engineering field linear or quadratic optimization problems are the exception. Insofar, it is questionable whether these types of problems are representative at all.

The evolution strategies show a linear or superlinear convergence behaviuor depending on the nature of the given problem. That means that the sequence of iteratively calculated design vectors $\mathbf{x}^{(k)}$ converges linearly or superlinearly towards the corresponding optimum solution $\mathbf{x}^*$, where the superscript k denotes a given k-th iteration step. We can mathematically express the convergence process by using the following relationship

$$\lim_{k\to\infty} \| \mathbf{x}^{(k)} - \mathbf{x}^* \| \to 0 \tag{7.1}$$

where the term $\| \mathbf{x}^{(k)} - \mathbf{x}^* \|$ represents an error norm (Euclidian norm). If the errors of subsequent iteration cycles are proportionally reduced such that

$$\frac{\| \mathbf{x}^{(k+1)} - \mathbf{x}^* \|}{\| \mathbf{x}^{(k)} - \mathbf{x}^* \|} = c_{lin} \tag{7.2}$$

where c_{lin} stands for a constant proportional factor, the convergence is said to be linear. This means that the number of accurate digits is constantly improved within the iteration steps. A quadratic convergence would be achieved if the number of accurate digits were doubled per iteration step. Hence, superlinear convergence is represented by a reduction of the error which lies between the linear and the quadratic convergence.

Although, quadratic convergence seems to be very attractive this type of convergence can be exclusively obtained by means of first and second partial derivatives of the objective function and the constraints (Newton methods, conjugate gradient methods and special variable metric methods). Despite the fact that the evaluation of derivatives is complicated and time-consuming, divergence difficulties are encountered if the optimization problem is not quadratic or not quasi-quadratic.

Consequently, if we demand to permit all potential types of nonlinearity, we have to relinquish methods which are confined to special categories of problems. For this reason, only methods with linear or superlinear convergence are of interest. Within the category of such methods, the evolution strategies are preferred to other popular methods because they are distinguished by the following essential features which are already mentioned above:

-improved reliability in complicated cases
-appropriate robustness in operation in the multidimensional case
-sufficient probability to also find global optima
-reasonable convergence rates
-insignificant storage demands
-advanced flexibility with changing problems

7.3. Theoretical concept of the evolution strategies

The evolution strategies can be divided into two main categories:

-two-membered (1+1)-strategy
-multi-membered $(\mu+\lambda)$ or (μ,λ)-strategies

Herein, the expression "members" means that competitive design vectors **x** are simultaneously stored and compared with respect to the given constraints and optimization criterion. The essential computational steps of the these strategies are to be described in the next four chapters. The individual steps are similar in all popular optimization techniques and can be summarized as follows:

- generation of new competitive design vectors
- selection of improved feasible vectors
- adjustment of the optimization parameters
- check upon the convergence and termination criteria.

The discussion of the theoretical concept has the purpose of providing the necessary information on the logic of the optimization process. (see also: [4],[5],[6],[10],[15],[16]). Along with the discussion, several further possible extensions of the theory will be outlined in order to demonstrate that the method can still be enhanced.

7.3.1. Generation process

7.3.1.1. (1+1)-strategy

In the case of the two-membered strategy two distinct vectors are considered, the currently best vector $\mathbf{x}^{(g)}$ and a new vector $\hat{\mathbf{x}}$ generated from the vector $\mathbf{x}^{(g)}$, both by means of a random process. Therefore, the self-explanatory notation (1+1) is used where each figure stands for a vector which competes with the other . The superscript (g) denotes the corresponding iteration cycle. The hat-symbol indicates that a check still has to be carried out to show whether an improved and feasible vector $\hat{\mathbf{x}}$ was obtained or not. The generation of a new vector $\hat{\mathbf{x}}$ is of crucial significance because the convergence behaviour is substantially affected by the logic of the generation. In general, we can state that the more powerful the variation mechanismi of the generation process are, the more efficient the convergence behaviour is. In the (1+1)-strategy the generation of new design vectors is governed by the simple vector equation

$$\hat{\mathbf{x}} = \mathbf{x}^{(g)} + \mathbf{z}^{(g)} = \mathbf{x}^{(g)} + \mathbf{R}\,\mathbf{s}^{(g)} \tag{7.3}$$

where the vector $\mathbf{z}^{(g)}$ is a random vector in which each of the n components $z_i^{(g)}$; $i=1,2,\ldots,n$ represents a Gauss-distributed step-length in the direction of the corresponding variable x_i. These step-lengths are generated by the random process

$$\mathbf{z}^{(g)} = \mathbf{R}\,\mathbf{s}^{(g)} \tag{7.4}$$

where $\mathbf{R}$ is a diagonal matrix called probability matrix,

$$\mathbf{R} = \begin{bmatrix} R_{11} & 0 & . & . & . & 0 \\ 0 & R_{22} & 0 & . & . & 0 \\ . & 0 & R_{33} & 0 & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{nn} \end{bmatrix} \tag{7.5}$$

and the vector $\mathbf{s}^{(g)}$ denotes the standard deviations of the step-length vector $\mathbf{z}^{(g)}$. The standard deviations represent the changes of the step-lenths and constitute important entities because they mainly influence the convergence rate of the iteration process.

To ensure a minimum of variation of the single variables x_i , $i=1,2,\ldots,n$, and to ensure that the last digit of an optimization variable has at least been varied, the following bounds are established:

$$s_i^{(g)} \geqslant \epsilon_{abs} \; ; \; i=1,2,3,\ldots,n \tag{7.6a}$$

and

$$s_i^{(g)} \geqslant \epsilon_{rel} \left| x_i^{(g)} \right| \tag{7.6b}$$

where ϵ_{abs} and ϵ_{rel} connotes the absolute and relative computer accuracy. ($\epsilon_{abs} \geqslant 0$ and $1+\epsilon_{rel} \geqslant 1$).

Each of the components R_{ii} , $i=1,2,\ldots,n$ is different and a standardized normally distributed parameter for which the Gauss distribution function $f(r)$ is valid. This function, written in terms of the general stochastic variable r, is defined by the following expression:

$$f(r) = \frac{1}{\sqrt{2\pi}}\, e^{-\frac{1}{2} r^2} \tag{7.7}$$

The corresponding distribution function $f(r)$ is a bell-shaped function and is shown in Fig. 7-1.

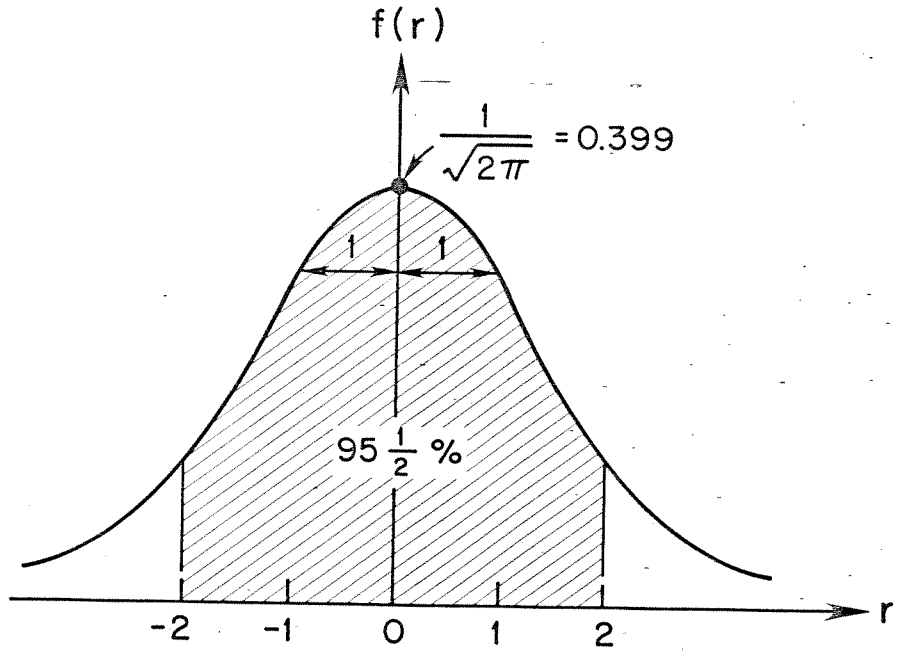

Fig.7-1: Standard normal distribution

It is important to note that the applied distribution function $f(r)$ generates great values of the parameter r less frequently, whereas small values of r are more probable. (This fact will have advantages if we want to approximate finite element equations during a potential re-analysis process to save computational effort. Due to the frequently small changes, we can approximate the corresponding finite element equations by Taylor-series expansions, without making major numerical errors.)

The Gauss normal distribution has been choosen because the generation of new points is considered as a mathematical simulation of the biological evolution process in which changes are also governed by normal distributions. Undoubtedly, the optimization mechanismi of the evolution process are very powerful and ,therefore , worth being simulated analagously within a mathematical optimization technique.

If we multiply the random parameter R_{ii}, for which the function $f(r)$ is valid, with the scalar $s_i^{(g)}$, according to the rules of statistic, we obtain a Gauss distributed parameter $z_i^{(g)}$ having the mean value 0 and the standard deviation $s_i^{(g)}$. The corresponding distribution function is given by

$$f(r\ s_i^{(g)}) = f(z_i^{(g)}) = f(R_{ii}\ s_i^{(g)}) = \frac{1}{\sqrt{2\pi}\, s_i^{(g)}}\ e^{-\frac{1}{2}\left[\frac{z_i^{(g)}}{s_i^{(g)}}\right]^2} \tag{7.8}$$

The next figure (Fig. 7-2) provides a graphical presentation of the bell-shaped Gauss distributions created.

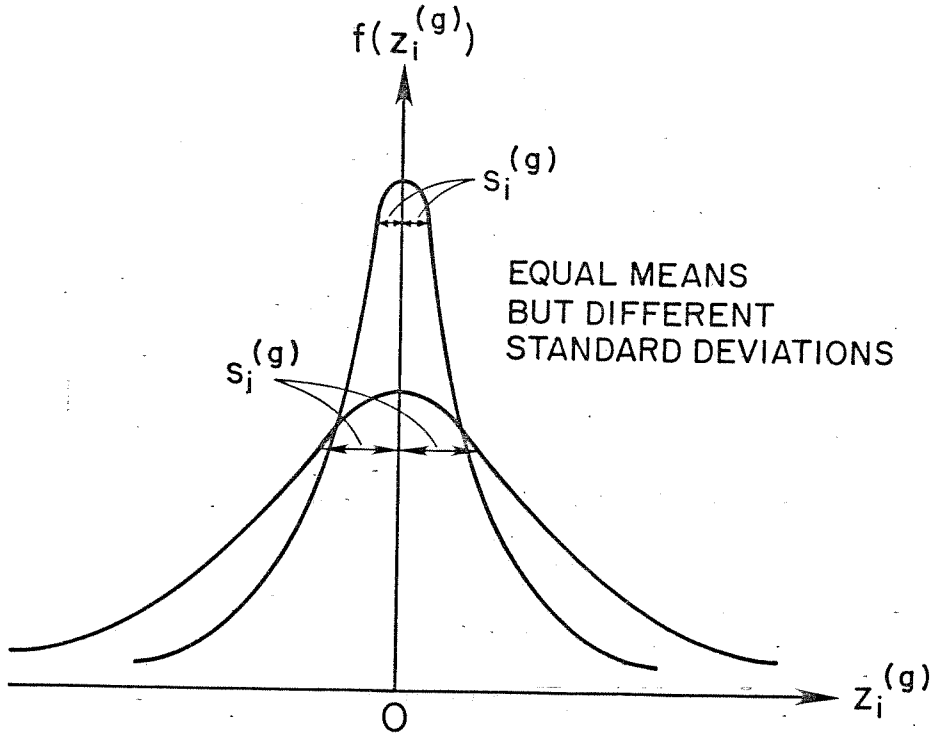

Fig.7-2: Gauss distribution

The product $R_{ii}\ s_i^{(g)}$ represents the step-length in the direction of the design variable x_i. It can be seen that the individual steps $z_i^{(g)}$ are all random and stochastically independent from each other because the matrix **R** is a diagonal matrix. Owing to this, the individual standard deviation $s_i^{(g)}$ only affects the step-lengths $z_i^{(g)}$, having the same index i, but not the other ones. Of course, we could also use a non-diagonal probability matrix which would correlate all or some of the step-lengths. However, the probabilistic model then involved would be inherently much more complex. Up till now, no appropriate theoretical solution could be found. Nevertheless, such a correlation would represent a further improvement of the optimization mechanism.

To illustrate the generation of $\hat{\mathbf{x}}$ vectors, for a two-dimensional case, a graphical impression of the vector equation (7.3) is given in Fig. 7-3 .

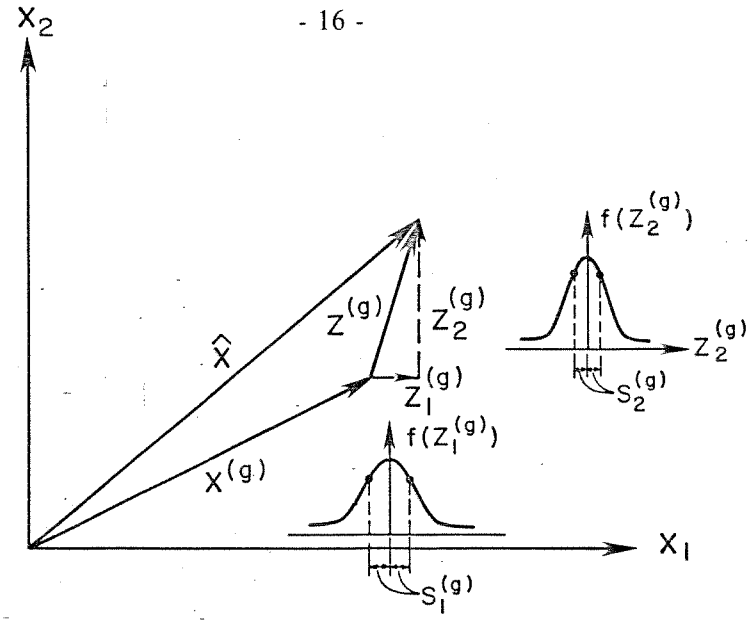

Fig.7-3: Random generation process

Due to the assumed stochastic independence of the individual step-lengths, we obtain a distribution function for the step-length vector which can be written as follows:

$$w(\mathbf{z}^{(g)}) = \prod_{i=1}^{i=n} w(z_i^{(g)}) = \frac{1}{(2\pi)^{\frac{n}{2}} \prod_{i=1}^{i=n} s_i^{(g)}} \, e^{-\frac{1}{2} \sum_{i=1}^{i=n} \left(\frac{z_i^{(g)}}{s_i^{(g)}} \right)^2} \tag{7.9}$$

By way of example, we can now calculate a first approximation for the mean value of the total length of the vector $\mathbf{z}^{(g)}$. For the sake of simplicity, let us assume that all standard deviations are equal to each other such that

$$s_i^{(g)} = s^{(g)} \; ; \; i=1,2,3,\ldots,n \tag{7.10}$$

This gives:

$$mean(|\mathbf{z}^{(g)}|) = s^{(g)} \sqrt{n} = s^{(g)} \, n^{1/2} \tag{10}$$

Accordingly, the most probable value for the total length of the vector $\mathbf{z}^{(g)}$ increases with respect to the square root of the dimension of the optimization problem. Vice versa, this result allows a very rough but useful estimation of the initial standard deviation s_{init}. If we replace the term $mean|\mathbf{z}^{(g)}|$ by the expected distance $\Delta\mathbf{x}$ between the starting vector and the (unknown) optimum, we obtain

$$s_{init} = \frac{\Delta \mathbf{x}}{\sqrt{n}} \tag{7.12}$$

This value can be used as a first approximation of the initial standard deviations. In addition, the variance can be calculated as well. We obtain:

$$var\,(|\mathbf{z}^{(g)}|) = 2 s^{(g)} n^{\frac{1}{4}} \tag{7.12a}$$

Equation (7.9) also indicates that the geometrical location of points which have the same probability is represented by a n-dimensional hyper-ellipsoid. This can be seen by evaluating the expression

$$\sum \left(\frac{z_i^{(g)}}{s_i^{(g)}} \right)^2 = const. \tag{7.13}$$

which obviously constitutes the equation of a n-dimensional hyper-ellipsoid.

Compared with other popular deterministic optimization techniques, the variation mechanism of the $(1+1)$-strategy is distinguished by the following characteristics:

-all directions in the n-dimensional space are equally probable, whereas the deterministic methods use fixed directions;
-the search step-lengths are Gauss distributed values which are characterized by small changes in the average, whereas deterministic methods are using search steps of arbitrary magnitude.

7.3.1.2. Multimembered $(\mu+\lambda)$ and (μ,λ)-strategies

In the case of multimembered methods a vector pool or set of new design vectors $\hat{\mathbf{x}}$ is generated. The governing equation can be written analogously to that of the $(1+1)$-strategy:

$$\hat{\mathbf{x}}_{\alpha\beta} = \mathbf{x}_\alpha^{(g)} + \mathbf{z}_{\alpha\beta}^{(g)} = \mathbf{x}_\alpha^{(g)} + \mathbf{R}\, \mathbf{s}_{\alpha\beta}^{(g)} \tag{7.14}$$

where the subscript $\alpha=1,2,...,\mu$ designates the vector $\mathbf{x}_\alpha^{(g)}$ which serves as one of the current basic vectors of the generation process. The subscript $\beta=1,2,...,\nu$ indicates the current number of the new vector generated from the basic vector $\mathbf{x}_\alpha^{(g)}$. Again, the superscript (g) is the iteration cycle counter, and the hat-symbol has the same connotation as described above.

The vector equation for the vector pool can also be written by using only one subscript:

$$\hat{\mathbf{x}}_\gamma = \mathbf{x}_\alpha^{(g)} + \mathbf{z}_\gamma^{(g)} \tag{7.15}$$

where

$$\gamma = 1,2,...,\nu,\nu+1,\nu+2,...,2\nu,2\nu+1,2\nu+2, \cdots ,\mu\nu \tag{7.16}$$

and

$$\mu\nu=\lambda \tag{7.17}$$

In the following we will refer to the notation which only needs one subscript.

Two sub-categories of multimembered strategies can be classified: If we consider both the basic points $\mathbf{x}_\alpha^{(g)}$ and the new points $\hat{\mathbf{x}}_{\alpha\beta}$ or $\hat{\mathbf{x}}_\gamma$, resp. , as potential competitors for the next iteration step $(g+1)$, a $(\mu+\lambda)$ - strategy can be developed. The $(\mu+\lambda)$-notation designates the fact that $\mu+\lambda$ vectors are grouped into the vector pool . To express this feature more precisely, let us introduce a new matrix $\mathbf{X}$ such that

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1\ \hat{\mathbf{x}}_2\ ,\ ...\ ,\ \hat{\mathbf{x}}_\lambda \mid \mathbf{x}_1^{(g)}\ \mathbf{x}_2^{(g)}\ ,\ ...\ ,\ \mathbf{x}_\mu^{(g)} \right] \tag{7.18}$$

where the matrix $\hat{\mathbf{X}}$ represents the vector pool and is ,therefore, called a "pool matrix". (Note that the components of the matrix $\mathbf{X}$ are vectors). However, we will limit the selection of points only to new points generated, even if basic points may represent better constellations. This decision has to do with the fact that we are trying to simulate the biological evolution process. The restriction of using only new points is to simulate the lethality mechanism in the biological evolution process. Hence,we will be considering only vector pools of the following type:

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1\ \hat{\mathbf{x}}_2\ \hat{\mathbf{x}}_3\ ,\ ...\ ,\ \hat{\mathbf{x}}_\lambda \right] \tag{7.19}$$

As can be seen, the vector pool exactly contains λ vectors which are generated from a vector set of μ vectors. To express this, the notation (μ,λ)-strategy is used where the comma is to

indicate that the μ basic vectors are not members of the vector pool $\hat{\mathbf{X}}$. The decision to perpetually reject basic points is based on numerical research. It can namely be proved that the convergence rate is not affected by basic vectors being better than succeeding ones if the number λ is a value greater than 5 μ .

The pool concept applied in the (μ,λ)- strategy couples the advantages of two different types of optimization strategies, sequential and simultaneous methods, by minimizing the disadvantages. Simultaneous methods (enumeration, Monto-Carlo-methods, etc.) improve the global convergence, however, they are time and memory consuming. Sequential methods (Newton, gradient and search methods) attempt to maximize the convergence rates, however, they are not reliable in multidimensional problems associated with multiple optima.

7.3.2. Selection process

7.3.2.1. (1+1)-strategy

In the (1+1)-strategy only one new design vector $\hat{\mathbf{x}}_\gamma$ exists. Also, only one basic vector $\mathbf{x}_\alpha^{(g)}$ is used. Thus, γ=1 and α=1, and we can relinquish the subscripts. The new design vector is only accepted as successor of the vector $\mathbf{x}_\alpha^{(g)}$ if it represents an improved value or at least an identical value of the optimization criterion and, of course, if it is admissible. This principle of rejection can be easily encoded.

In order to have a consistent formal notation for both types of strategies, a matrix formulation is to be used. According to the $(\mu+\lambda)$ or (μ,λ)-notation, we obtain a pool matrix into which only two vectors are grouped.

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}} \mid \mathbf{x}^{(g)}\right] \tag{7.20}$$

The vector $\mathbf{x}^{(g+1)}$ of the next iteration cycle can be understood as the result of a matrix multiplication such that

$$\mathbf{x}^{(g+1)} = \hat{\mathbf{X}}\,\hat{\mathbf{B}}\,\mathbf{b}_1 \tag{7.21}$$

where $\hat{\mathbf{B}}$ is a Boolean matrix and $\mathbf{b}_1$ is a unit vector. With respect to the feasible region S^n the matrix $\mathbf{B}$ is defined as follows

$$\hat{\mathbf{B}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \text{ if } f(\hat{\mathbf{x}}) \leqslant f(\mathbf{x}^{(g)}) \;;\; \hat{\mathbf{x}}, \mathbf{x}^{(g)} \in S^n \tag{7.22}$$

or

$$\hat{\mathbf{B}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \text{ if } f(\hat{\mathbf{x}}) > f(\mathbf{x}^{(g)}) \tag{7.23}$$

The vector $\mathbf{b}_1$ is a vector in which the first component is equal to 1, while the second is 0. Thus

$$\mathbf{b}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{7.24}$$

If the first Boolean matrix holds, the new vector $\hat{\mathbf{x}}$ is accepted as the successor. If not the old vector $\mathbf{x}^{(g)}$ is taken. Since the matrix multiplication only has a formal significance, the evaluation is not necessary. Rather, the rejection principle, being more rapid than a time consuming matrix multiplication, can be encoded by a fundamental condition statement.

7.3.2.2. (μ,λ)-strategy

In the (μ,λ)-strategy, we have a pool matrix which contains λ vectors. In order to obtain a code which can be easily programmed ,and to avoid a code which is too memory consuming, we decide on how many of the vectors of the pool are to be selected as basis vectors during the next iteration cycle $(g+1)$. We demand the selection of the μ best vectors of a total of $\mu\nu=\lambda$ potential vectors. This restriction allows us to make repetitve use of the vector generation-equation described above. Such a proceeding results in a constant number of basic vectors $\mathbf{x}_\alpha$, $\alpha=1,2,...,\mu$. (constant population size). Of course, a variable number of basic vectors would also be possible (e.g. during a phase of weak convergence to improve the convergence rate) ,however, that would lead to a more complex approach. (Development in progress.)

The selection of the μ best vectors can be also described by the following matrix equation

$$\overline{\mathbf{X}} = \hat{\mathbf{X}}\,\hat{\mathbf{B}} \tag{7.25}$$

where $\hat{\mathbf{B}}$ is a Boolean matrix of type (λ,μ). In this matrix all elements are 0 except those elements which have the value of 1, and whose row and column numbers represent the position number of one of the μ best vectors $\hat{\mathbf{x}}_\gamma$ in the pool matrix $\mathbf{X}$. To elucidate the selection in more detail, let us consider the following example

$$\mu=2 \quad \nu=3 \quad \lambda=\mu\nu=6 \tag{7.26}$$

Thus, the pool matrix would be

$$\hat{\mathbf{X}} = \left[\hat{\mathbf{x}}_1 \;\; \hat{\mathbf{x}}_2 \;\; \hat{\mathbf{x}}_3 \;\; \hat{\mathbf{x}}_4 \;\; \hat{\mathbf{x}}_5 \;\; \hat{\mathbf{x}}_6\right] \tag{7.27}$$

Providing that the vectors $\hat{\mathbf{x}}_3$ and $\hat{\mathbf{x}}_4$ represent the best competitors within the pool, the position numbers are 3 and 4 , respectively. As a consequence, the Boolean matrix $\mathbf{B}$ must be written as

$$\hat{\mathbf{B}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7.28}$$

Hence,

$$\overline{\mathbf{X}} = \hat{\mathbf{X}}\,\hat{\mathbf{B}} = \left[\hat{\mathbf{x}}_3 \;\; \hat{\mathbf{x}}_4\right] \tag{7.29}$$

The individual basic vectors can be derived from the matrix $\overline{\mathbf{X}}$ by post-multiplication with Boolean vectors $\mathbf{b}_\alpha$, $\alpha=1,2,...,\mu$. The subscript α indicates the row in which the element has the value of 1. All other elements are 0 by definition.

For our example mentioned above we would obtain

$$\mathbf{b}_1 = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \tag{7.30a}$$

or, respectively

$$\mathbf{b}_2 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \tag{7.30b}$$

Hence

$$\mathbf{x}_1^{(g+1)} = \overline{\mathbf{X}}\,\mathbf{b}_1 = \hat{\mathbf{x}}_3 \; ; \; \mathbf{x}_2^{(g+1)} = \overline{\mathbf{X}}\,\mathbf{b}_2 = \hat{\mathbf{x}}_4 \tag{7.31}$$

Consequently, we can describe the selection process in accordance with the formulation already applied in the (1+1) - strategy. (See equation (7.21)).

$$\mathbf{x}_{\alpha}^{(g+1)} = \hat{\mathbf{X}}\,\hat{\mathbf{B}}\,\mathbf{b}_{\alpha} \tag{7.32}$$

Again, it should be emphasized that the applied Boolean matrices have only formal significance and are not needed in the computer program creation because they can be replaced by more efficient and elementary statements having the same effect.

In comparision with the (1+1)-strategy the (μ,λ)-strategy gives improved and enhanced mechanismi which provide an improved reliability. The run-time, by contrast, must be expected to increase due to the multiple vector set. However, it is essential to note that the numerical effort does not proportionally increase with respect to the number of vectors per set, as one could perhaps expect. The reason for that lies in the more powerful optimization mechanismi incorporated which improve the convergence. In addition, it should be mentioned that computer time will be tremendously reduced if one can make use of advanced computer facilities, like modern parallel or pipeline processors. Due to the parallel logic of the (μ,λ)-stategy, the corrsponding algorithms are well suited for implementation with respect to such types of computers. Therefore, the multimembered strategy constitutes a numerical method which has a further significant potential, not yet numerically materialized.

7.3.3. Adjustment of the optimization process

7.3.3.1. (1+1)-strategy

In order to achieve a rapid convergence it is obvious that the step-lengths, or their corresponding standard deviations, must have the appropriate size. The next figure (Fig. 7-4) illustrates this demand for a two-dimensional situation with the design variables x_1 and x_2 and a smooth function representing a gradually ascending ridge .

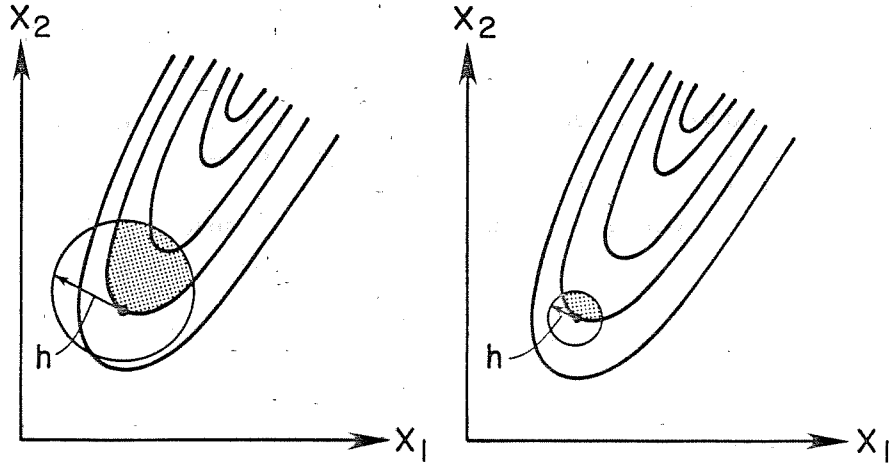

Fig.7-4: Effect of step-sizes

For the sake of simplicity, let us assume that all random step-lengths $\mathbf{z}^{(g)}$ may be generated on

a cycle with the radius h. If h is too great the effectiveness of the search is poor, because although one success leads to a great advance , relatively few trials will be successful. (see dotted portion of the cycle). On the other hand, if h is too small the effectiveness is also low, because although a higher proportion of trials will be successful, the absolute movement will be small. Obviously , there must be an optimal values for h that guarantuees an optimal efficiency.

In order to find such an optimal step-length during the optimization, the ratio of number of successes to total number of trials may be used as a measure of convergence. This ratio represents the probability of effectiveness and can be called success probability.

$$w_e = \frac{number\ of\ successful\ trials}{total\ numbers\ of\ trials} \tag{7.33}$$

At first glance, it seems impossible to specify a value for w_e which may lead to convergence for every arbitary optimization criterion. However, it can be proved that this ratio is almost independent of the nature of the objective criterion. By regarding two special functions, both of which represent diametrically opposite optimization functions within a wide range of potential functions,it could be shown that the optimum value of w_e, guaranteeing the best adjusted step-lengths, varies only between $\frac{1}{3.8}$ and $\frac{1}{5.4}$. Therefore, an approximate value $1/5$ was choosen. That means, if

$$w_e = \frac{1}{5} \tag{7.34}$$

the step-lengths are optimal and the assumed standard deviations are correctly dimensioned. If the ratio w_e is less than $1/5$ the step-vector $\mathbf{z}^{(g)}$ is to be reduced , if w_e is greater than $1/5$ the vector is to be increased.

The magnitude of the reduction or increment is based on a theoretical approach for the special function that constitutes the inferior case. Theoretical calculations resulted in the fact that all standard deviations must be reduced with the factor $\frac{1}{1.2}$ if w_e is less than $1/5$, and must be increased with the factor 1.2 if w_e is greater than $1/5$. It should be mentioned that the $1/5$ -rule used in the $(1+1)$-strategy makes no assertion about the relative proportions of the single standard deviations. Thus all elements of the vector $\mathbf{z}^{(g)}$ are reduced or incremented at the same time. In the multimembered strategies improved in-built mechanismi provide the possibility to adjust the relations of the individual elements as well.

To measure the success probability the average value of w_e through several trails is to be calculated. For numerical purposes, the evaluation of the $1/5$ - rule is performed such that, after a number of $10n$ trails, the number of successes is checked. Then, it is checked how many successes have been encountered during the last $10n$. Obviously,an amount of successful trials less than $2n$ corresponds to a value of w_e which is less than $\frac{2n}{10n} = \frac{1}{5}$. In this case, the step-lengths must be reduced. If the amount is greater than $2n$ the step-lengths must be increased.

One particular difficulty is to be mentioned : The $1/5$ -rule assumes that there is always a combination of standard deviations by which, on an average , at least one improvement of the optimization criterion within 5 trials is to be expected. However, constellations are conceivable at which this condition is not fulfilled at all times. Such a pathological case may occur on certain parts of the boundary where several constraints may be active, with the consequence that the success probability is continously overestimated. Therefore, a premature halt not desired could take place.

This type of phenonema is demonstrated in the following example (see Fig. 7-5).

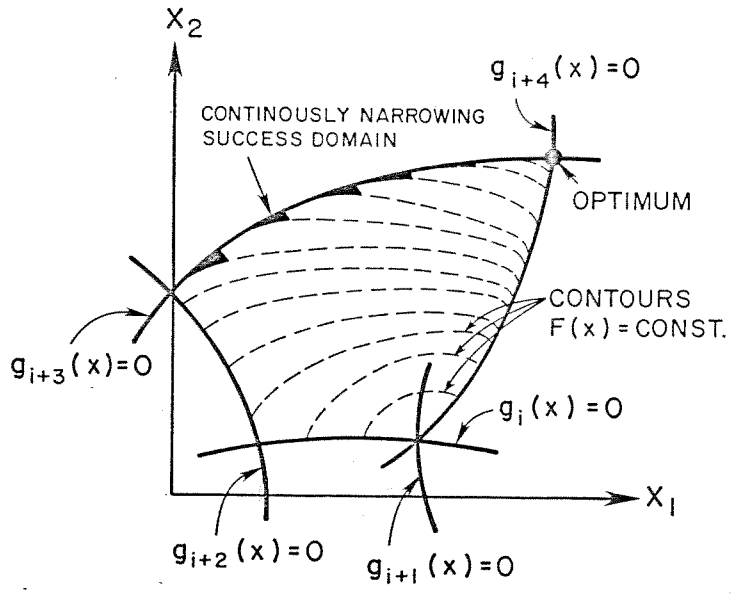

Fig.7-5: Pathological constellation

Using the 1/5 - rule,this example would result in a rapid reduction of the step-lenth because it is impossible to obtain 1 success within 5 trials, in the average. The reason for this is that there is only a very small domain of success (filled-in sector of the cycle). The present case is inherently much more sensitive than the general case represented by the 1/5 - rule. Therefore, it is a good idea to modify the general 1/5 -rule in such a way that a greater value than 5 is used. (In the corresponding subroutine this can be accomplished by increasing the default value of a special control parameter.)

Such types of numerical instability are problem dependent, and not a pecularity of the numerical optimzation routine applied. Actually, in other popular methods numerical difficulties are also encountered. (They are, however, neither mentioned nor are there possibilities to handle such instabilities.) To overcome such instabilities, one could replace the current optimization criterion by an augmented objective criterion, using a Lagrangian or penalty formulation. (The active constraints are temporarily implemented into the augmented optimzation criterion and ,therefore, inactivated). A further way is to employ multimembered evolution strategies which are more powerful in pathological cases.

7.3.3.2. (μ,λ)-strategy

The (1,1)-strategy provides an adjustment of step-lengths which is similar to that of the popular deterministic optimization strategies. By contrast, the (μ,λ)-strategy provides an adjustment-mechanism essentially different. Adjustments are performed analogously to the biological evolution process.

To achieve maximal convergence rates, the standard deviations are additionally incorporated into the random variation and selection process, just as well as the design or optimization variables. That is to say that a design constellation is represented not only by its design variables but also by its standard deviations. Of course, the decision of whether a new design vector $\hat{\mathbf{x}}$ will be accepted as a basis of the next iteration cycle $(g+1)$ solely depends on the design variables, and not on the values of the standard deviations. However, the magnitude and nature of the succeeding changes of the design variables is influenced by the random nature of the standard deviations associated with these design variables. Basically, the design or optimization variables constitute object parameters whereas variable standard deviations can be understood as strategy parameters. Both types of parameters are subjected to the random control mechanism. Therefore, the 1/5-success-rule used in the (1,1)-strategy is no longer

necessary because the (μ,λ)-logic contains a natural success-rule enbodied in it.

However, the 1/5 -rule is replaced by a new rule which requires that the ratio of the number of the basic vectors μ and the number of new vectors λ generated from them must have the right proportion, namely such that

$$\frac{\lambda}{\mu} > 5 \tag{7.35}$$

While the $\frac{\lambda}{\mu}$-rule is fullfilled, eventually better basic design vectors $\mathbf{x}_\alpha^{(g)}$, $\alpha=1,2,3,...,\mu$, have no effect on the convergence. Therefore, we prefer the (μ,λ)-strategy to the $(\mu+\lambda)$-strategy because we can save memory storage. (To a certain extend, that decision already constitutes a portion of the adjustment principle).

Along with the random generation of new basic vectors $\mathbf{x}_\alpha^{(g)}$ the standard deviations are also randomly generated. This random generation is governed by the following vector equation:

$$\mathbf{s}_\alpha^{(g+1)} = \mathbf{Z}\, \mathbf{s}_\alpha^{(g)} \tag{7.36}$$

where $\mathbf{s}_\alpha^{(g+1)}$ designates the vector of the standard deviations of the μ best design vectors $\mathbf{x}_\alpha^{(g+1)}$ within the next generation $(g+1)$. The matrix $\mathbf{Z}$ is a diagonal matrix containing random diagonal elements. Thus

$$\mathbf{Z} = \begin{bmatrix} Z_{11} & 0 & . & . & . & 0 \\ 0 & Z_{22} & 0 & . & . & 0 \\ . & 0 & Z_{33} & 0 & . & 0 \\ . & . & . & . & . & 0 \\ 0 & 0 & 0 & 0 & 0 & Z_{nn} \end{bmatrix} \tag{7.37}$$

The single elements Z_{ii} , $ii=1,2,3,...,n$ are logarithmically normally distributed parameters. This type of distribution function is required because the random values of Z_{ii} must have the mean value 1 , due to the multiplication process, and increments and reductions. Also, increments and reductions generated need to be equally probable, and additionally, great changes of Z_{ii} values are to be less frequently than small ones. The distribution function satisfying these three requirements is a logarithmic normal distribution which can be written in terms of the general variable ζ as follows:

$$f(\zeta) = f(Z_{ii}) = \frac{1}{\sqrt{2\pi}\,\bar{\sigma}}\,\frac{1}{\zeta}\, e^{-\frac{1}{2}\left[\frac{\ln\zeta}{\bar{\sigma}}\right]^2} \tag{7.38}$$

where $\bar{\sigma}$ is an unknown parameter which represents the changes of the standard deviations (standard deviation of the standard deviation). Although it would be possible to consider individual $\bar{\sigma}$ parameters for each of the subscripts i , $i=1,2,3,...,n$, up till now predictions of convergence can only be made for the particular case of one parameter $\bar{\sigma}$ which is common for all subscripts i. In this elementary case, based on the theory of probability, an appropriate estimation of $\bar{\sigma}$ could be found. (In general, the parameter $\bar{\sigma}$ depends on the number λ and the current topology of the optimization space). Let us consider a representative example: it is a good idea to set the unknown parameter $\bar{\sigma}$ equal to 1 for a (10,100)-strategy, and to increment this value sublinearly. The generation of ζ or Z_{ii}-values, respectively, can be readily encoded by using $(0,\bar{\sigma})$ normally distributed parameters, say ξ, and by evaluating the exponential expression e^{ξ}. Thus

$$Z_{ii} = \zeta = e^{\xi} \tag{7.39}$$

Using a vector pool provides another exceptionally powerful optimization mechanism called scaling or recombination of the individual optimization parameters. Recombination or scaling means the generation of new design vectors through combination of the individual design variables x_i , $i=1,2,3,...,n$ which are associated with the basic vectors

$\mathbf{x}_\alpha^{(g)}$, $\alpha=1,2,3,...,\mu$. The recombination mechanism for a particular variable, say $\tilde{x}_i$, can be mathematically described by means of a scalar product of two vectors. The first vector (row vector) contains all those elements of the basic vector $\mathbf{x}_\alpha^{(g)}$ having the same subscript i. If this vector is denoted $\mathbf{X}_i^T$ we have:

$$\tilde{\mathbf{X}}_i^T = \left\{ x_{1,i}^{(g)} \; ; \; x_{2,i}^{(g)} \; ;...; \; x_{\mu,i}^{(g)} \right\} \tag{7.40}$$

The second vector (column vector), denoted $\mathbf{b}_\kappa$, is a Boolean vector with μ-elements all of which are "0", except for one which has the value "1". The position number κ of the value "1" is randomly determined, where all positions $\kappa=1,2,3,...,\mu$ are equally probable. Hence, a particular optimization variable $\tilde{x}_i$ recombined can be written as

$$\tilde{x}_i = \tilde{\mathbf{X}}_i^T \mathbf{b}_\kappa \tag{7.41a}$$

If the above scalar product is carried out for all subscripts i , $i=1,2,3,...,n$, a new recombined vector $\tilde{\mathbf{x}}_\alpha$ is created.

$$\tilde{\mathbf{x}}_\alpha = \left\{ \begin{matrix} \tilde{x}_1 \\ \tilde{x}_2 \\ . \\ . \\ \tilde{x}_n \end{matrix} \right\} ; \; \alpha=1,2,3,...,\mu \tag{7.41b}$$

Each recombined vector is then subjected to the regular random generation process already previously described. In the same fashion the standard deviations can be recombined. Accordingly, we obtain

$$\tilde{\mathbf{s}}_i = \tilde{\mathbf{S}}_i^T \mathbf{b}_\kappa \tag{7.41c}$$

and

$$\tilde{\mathbf{s}}_\alpha = \left\{ \begin{matrix} \tilde{s}_1 \\ \tilde{s}_2 \\ . \\ . \\ \tilde{s}_n \end{matrix} \right\} ; \; \alpha=1,2,3,...,\mu \tag{7.41d}$$

Finally, it should be emphasized that the recombination process is of substantial significance with respect to global convergence and reliability. Solely by means of the recombination process, appropriately adjusted step-lengths can be achieved, especially in problems with an intricate topology. However, even though the (μ,λ)-strategy incorporates advanced mechanismi, nonlinear optimization problems with very intricate topologies must be carefully treated. One always has to check the obtained results with regard to plausibility.

7.3.4. Convergence and termination criteria

The optimization is terminated if the CPU-time has elapsed, or if the the optimum is found with respect to the required convergence tolerances.

7.3.4.1. (1+1)-strategy

Convergence is assumed if one of the two required tolerances is achieved

$$f(\mathbf{x}^{(g-\Delta g)})-f(\mathbf{x}^{(g)})\leqslant\epsilon_{abs} \tag{7.42}$$

where ϵ_{abs} is the absolute tolerance and Δg indicates the intervall in which the convergence is to be checked. The relative convergence is checked by means of the criterion

$$f(\mathbf{x}^{(g-\Delta g)})-f(\mathbf{x}^{(g)})\leqslant\epsilon_{rel}\,|f(\mathbf{x}^{(g)})| \tag{7.43}$$

where ϵ_{rel} denotes the relative convergence tolerance desired. The retarded check upon the convergence accuracy results in a damping effect with the purpose of avoiding a premature halt in cases where the step-lengths have to be rapidly changed. In the computer program creation it is assumed that

$$\Delta g=20\,n \tag{7.44}$$

to secure sufficient increments or reductions of the step-lenths within the current Δg-intervall. (The corresponding subroutine also provides the possibility to change the Δg-intervall after which a check of the convergence is carried out.)

7.3.4.2. (μ,λ)-strategy

The criteria of the (1+1)-strategy are slightly modified. Convergence is assumed if the values of the optimization criteria associated with the vectors $\mathbf{x}_\alpha^{(g)}$ are close to one another with regard to the absolute and the relative accuracy, ϵ_{abs} and ϵ_{rel} ,respectively. This demand can be mathematically described by means of the difference between the best and the worst value of the optimization criterion within the current set of basic vectors considered. Thus

$$\underset{\alpha}{Min}\left[\,f(\mathbf{x}_\alpha^{(g)})\,\right]-\underset{\alpha}{Max}\left[\,f(\mathbf{x}_\alpha^{(g)})\,\right]=Min\ f-Max\ f\leqslant\epsilon_{abs} \tag{7.45}$$

where

$$\alpha=1,2,3,\ldots,\mu \tag{7.46}$$

The relative convergence ϵ_{rel} is checked by way of the corresponding relative criterion as follows :

$$Min\ f-Max\ f\leqslant\frac{\epsilon_{rel}}{\mu}\sum_{\alpha=1}^{\alpha=\mu}f(\mathbf{x}_\alpha^{(g)}) \tag{7.47}$$

For further details it is referred to the report [16]. In this report the computer implementation of the evolution strategies, and selected numerical examples are dealt with.

8. Re-analysis formulation of the finite element model

8.1. Necessity

During the optimization process, data of the structural response are perpetually needed for repetitive evaluation of the structural constraints and, conditionally, for calculation of the objective criterion. To update these data with respect to optimization variables, actually, a complete re-run of the finite element program is required. As we know, however , the cost for complete re-runs can be considerable. In particular, if systems with a major number of mechanical degrees of freedom are to be optimized, the multiple re-analysis process can be time consuming in such a degree that the complete analysis must be approximated, otherwise, the structural optimization can become prohibitive.

8.2. Numerical effort of the total structural optimization model

Despite the category of the applied optimization algorithm, the total numerical effort of a given structural optimization problem of n optimization variables x_i , i=1,2,3,...,n can be estimated by the following relationship

$$E_{TOT} = n^2 \, (E_{OBJ}+E_{CON}+E_{FEM}+E_{EIG}) \tag{8.1}$$

which can be readily derived from the expression in equation (6.2). There, it was shown that the computational effort of all optimization procedures currently available is at least governed by numerical operations of order $O(n^{2+FAK})$. Herein, the portion $O(n^{FAK})$ represented the average effort needed to evaluate the objective function. In structural optimization problems, this portion must be replaced by the term $(E_{OBJ}+E_{CON}+E_{FEM}+E_{EIG})$ where

E_{OBJ} = numerical effort to evaluate the present objective criterion, according to relationship (5.1)

E_{CON} = numerical effort to evaluate the set of m constraints, according to relation ship (5.1), provided that the numerical effort for the finite element analysis is completely represented by the term E_{FEM}

E_{FEM} = numerical effort to analyse the structural response and all quantities required for the structural design (e.g. stresses, forces, etc.), according to relationships (5.1) and (5.2), respectively

E_{EIG} = numerical effort to solve the involved eigenvalue problem, according to relationships (5.1) and (5.3), respectively

For the present estimation of effort, however, we will only focuse on the conventional equilibrium field problem represented by the equation set (5.2), and the denotations "analysis" and "re-analysis" are exclusively related to that category of problems. In other words, until further notice, the repetitive solution of eigenproblems is not considered. Rather it is assumed that eigenproblems may be solved through more elementary approaches. Such a simplification often leads to reasonable results unless very complex structural systems (spatial frames or mixed types of elements) are to be optimized. Therefore, the assumption is made that the numerical effort E_{EIG} is represented along with the effort E_{CON} needed to calculate constraints. Based on that assumption some very important conclusions can be drawn from equation (8.1) :

(1) Due to the term n^2 every savings within the finite element analysis as well as the evaluation of the constraints and objective criterion improves the effectivity of the overall problem

(2) The optimization process is dominated by the finite element analysis if

$$E_{FEM} \geqslant E_{OBJ}+E_{CON} \tag{8.2}$$

(3) The evaluation of the constraints is more time-consuming than the evaluation of the objective function if

$$E_{CON} \geqslant E_{OBJ.} \tag{8.3}$$

Even though the evaluation of structural constraints is directly coupled with the repetitive finite element analysis, for the sake of an easier definition, we consider the effort for the finite element analysis and the structural constraints as decoupled entities. The advantage of such a decoupling is that we can approximate the three above quantities E_{OBJ} , E_{CON} and E_{FEM} in terms of the main characteristic parameters of the individual computational models associated with the structural optimization process. As far as the numerical optimization is concerned, these parameters are the total number n and m of the optimization variables and constraints, respectively. As far as the finite element analysis is concerned these parameters are the total number of degrees of freedom N_{dof}, and the mean bandwidth of the global stiffness matrix B_{mean}, supposing that the structure is subjected to only one loading.

With regard to the comments already made in paragraph 6, the first term E_{OBJ} in equation (8.1) can be expressed by the following formula:

$$E_{OBJ} = n^{FAK_1} \tag{8.4}$$

where

FAK_1 = factor which represents the effort needed to calculate the value of the objective criterion; in elementary cases we can set FAK_1 approximately equal to values not greater than 1.

The mean effort needed to evaluate the individual constraints, can be expected to be not greater than that required to calculate the objective criterion. Hence, the second term in equation (8.1), which is the effort for evaluating all m constraints, is represented by

$$E_{CON} = m\ n^{FAK_2};\ FAK_2 \leqslant FAK_1 \tag{8.5}$$

where

FAK_2 = factor which represents the average effort to calculate one of the m constraints; usually, FAK_2 varies in the range between 1/2 and 1.

The third term E_{FEM}, which represents the numerical effort of a complete finite element analysis, depends on the effectivity of the

- input
- assemblage
- load vector generation
- triangularization (provided that elimination methods are applied)
- back-substitution (provided that elimination methods are applied)
- calculation of subsequent structural quantities
- output.

For our purposes, however, it is sufficient to make use of a very rough approximation which is exclusively based on the phase of the solution of stiffness equations. (triangularization and back-substitution). Through this simplification the total effort of a complete finite element analysis E_{FEM} can approximately be predicted. Providing that highly efficient equation solvers are applied (like those devoloped by WILSON et al. [17],[18]) we can estimate the term E_{FEM} as follows:

$$E_{FEM} \approx 3 \frac{N_{dof}\ B_{mean}^2}{2} \tag{8.6}$$

where

N_{dof} total number of degrees of freedom

B_{mean} mean bandwidth of the global stiffness matrix

3 approximation factor by means of which the partial effort needed for the solution phase ($\approx \frac{N_{dof}\ B_{mean}^2}{2}$) must be multiplied to approximate the effort E_{FEM}.

Since we are only considering linear elastic and discrete structures, additionally, the mean bandwidth B_{mean} can be roughly approximated by

$$B_{mean} \approx \frac{2}{10} N_{dof} \tag{8.7}$$

Consequently, if the finite element analysis is completely carried out, the optimization would be dominated by the finite element phase when

$$N_{dof} \geqslant (\frac{100}{6})^{\frac{1}{3}} (n^{FAK_1} + m\ n^{FAK_2})^{\frac{1}{3}} \tag{8.8}$$

To give a numerical example let us assume that

$$n = 100\ ;\ m = 100\ ;\ FAK_1 \approx 1\ ;\ FAK_2 \approx \frac{1}{2} \tag{8.9}$$

Consequently, if the number of degrees of freedom already has a value of

$$N_{dof} \geqslant (\frac{100}{6})^{\frac{1}{3}} (n + m\sqrt{n})^{\frac{1}{3}} \approx 27 \tag{8.10}$$

the finite element analysis would dominate the optimization process. Also, the evaluation of the constraints would need more effort than that of the ojective criterion because, according to equation (8.3),

$$m \geqslant n^{FAK_1} = \sqrt{n} = 10 \tag{8.11}$$

Obviously, the above numerical example demonstrates that

(a) the finite element analysis dominates the numerical effort during the optimization process if structures with already minor degrees of freedom occur.

(b) great numbers of constraints have an essential effect on the numerical effort.

8.3. Approximation schemes at overall system level

Within the last, years several methods for avoiding complete re-analysis after a design change have been proposed. In most cases these methods are based on power series expansions [19], [20], [21]. More recently, methods based on rational approximation [22] have been reported.

It can be observed that methods based on power series expansions may suffer from major convergence difficulties when a larger change of the optimization or design variables x_i , $i=1,2,3,...,n$ is created through the applied optimization algorithm. On the contrary, these methods are very effective for small design changes. This property occurs because the domain of convergence of a given power series may be restricted to a circular region of finite radius. As a consequence, a bound on the size of possible design changes is established.

For that reason rational approximants have been proposed, which are not subjected to such particular limitations. Therefore, rational approximants may be more generally applicable than approximations based on power series expansions. However, although they may be very promising, some substantial difficulties and questions still remain at present:

(a) since the method is theoretically capable of handling divergence sequences of solutions, numerical overflows may often occur,

(b) experience with respect to a major number of optimization variables is missed.

By contrast, the latter difficulties are not encountered if approximants based on power series expansions are used. In addition, the divergence behaviour observed in the case of major changes of the optimization variables can be compensated if power series are associated with evolution strategies. As stated in chapter 7.3 (generation process of new design vectors), the evolution strategies are mainly distinguished by small changes of the optimization variables due to Gauss-distributed search step-lengths. Therefore, the approximation error made by using power series expansions is vastly minimized. That is to say that the evolution strategies obviously represent optimization strategies appropriate with respect to re-analysis based on power series. Hence, in the following, power series expansions (Taylor series) will be applied as approximants.

Using Taylor series permits to transform the originally implicit re-analysis problem, according to equations (5.1), (5.2), (5.3) and (5.4), into an explicit formulation. In particular, the structural displacement response can be expressed as an explicit function in terms of the optimization vector $\mathbf{x}$. However, explicit expressions can be only acquired at the expense of additional iterations to be performed within the iterative optimization process. The computation of explicit functions is a straightforward procedure, and will be illustrated in the followings.

8.3.1. Linear finite element equations

Let us assume that, in general, the current stage of the optimization process may be represented by the optimization vector $\mathbf{x}^{(g)}$. (Also, see equations (7.3) and (7.14) by which current vectors are generated). For example, this vector might be based on previous experience with similar structures, or might be a feasible solution created during the optimization process. Provided that no eigenvalue problem is involved, we have to solve the following system of linear equations:

$$\mathbf{K}(\mathbf{x}=\mathbf{x}^{(g)}) \bullet \mathbf{u}(\mathbf{x}=\mathbf{x}^{(g)}) = \mathbf{P}(\mathbf{x}=\mathbf{x}^{(g)}) \tag{8.12}$$

or shorter

$$\mathbf{K}_g \, \mathbf{u}_g = \mathbf{P}_g \tag{8.13}$$

where the subscript g identifies the situation at $\mathbf{x}=\mathbf{x}^{(g)}$. Subsequently, if the design vector $\mathbf{x}^{(g)}$ is modified with respect to a change $\Delta\mathbf{x}$, a new stiffness matrix $\mathbf{K}(\mathbf{x}^{(g)}+\Delta\mathbf{x})$ (stiffness modification) ought to be created. If we would apply the terminology of the evolution strategies, the change $\Delta\mathbf{x}$ would be identical to the terms $\mathbf{z}^{(g)}$ and $\mathbf{z}_{\alpha\beta}^{(g)}$, respectively. (see eqn. (7.3) and (7.14)).

However, to avoid solving the corresponding response $\mathbf{u}(\mathbf{x}^{(g)}+\Delta\mathbf{x})$ in

$$\mathbf{K}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) \bullet \mathbf{u}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) = \mathbf{P}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) \tag{8.14}$$

or shorter

$$\mathbf{K}(\hat{\mathbf{x}}) \bullet \mathbf{u}(\hat{\mathbf{x}}) = \hat{\mathbf{K}}\,\hat{\mathbf{u}} = \hat{\mathbf{P}} = \mathbf{P}(\hat{\mathbf{x}}) \tag{8.15}$$

we make use of the re-analysis technique, where the hat-symbol identifies the situation at $\mathbf{x}^{(g)}+\Delta\mathbf{x}$. Since the elements of the matrix $\mathbf{K}$, and the vectors $\hat{\mathbf{u}}$ and $\mathbf{P}$ are regarded as continuously differentiable, we can expand all quantities as matrix or vector-valued power series around the current design vector $\mathbf{x}^{(g)}$. Thus, using Taylors's theorem and neglecting cubic terms and higher ones, we can approximate the global stiffness matrix of the updated design as follows:

$$\hat{\mathbf{K}} = \mathbf{K}_g+\Delta\mathbf{K} \approx \mathbf{K}_g+\frac{\partial\mathbf{K}}{\partial\mathbf{x}}\Delta\mathbf{x}+\frac{1}{2}\Delta\mathbf{x}^T\frac{\partial^2\mathbf{K}}{\partial\mathbf{x}\,\partial\mathbf{x}}\Delta\mathbf{x}+\cdots=\mathbf{K}_g+\delta\mathbf{K}_g+\frac{1}{2}\delta^2\mathbf{K}_g+\cdots \tag{8.16}$$

where

$\Delta\mathbf{K}$ = contribution to the global stiffness matrix due to the current design change $\Delta\mathbf{x}$.

Correspondingly, we can expand the displacement respone $\hat{\mathbf{u}}$ by Taylor series

$$\hat{\mathbf{u}} = \mathbf{u}_g + \Delta\mathbf{u} \approx \mathbf{u}_g + \frac{\partial \mathbf{u}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \frac{\partial^2 \mathbf{u}}{\partial \mathbf{x}\,\partial \mathbf{x}}\Delta\mathbf{x} + \cdots = \mathbf{u}_g + \delta\mathbf{u}_g + \frac{1}{2}\delta^2\mathbf{u}_g + \cdots \tag{8.17}$$

where

$\Delta\mathbf{u}$ = contribution to the overall displacement vector due to the design change $\Delta\mathbf{x}$.

Furthermore, for the load vector we have

$$\hat{\mathbf{P}} = \mathbf{P}_g + \Delta\mathbf{P} \approx \mathbf{P}_g + \frac{\partial \mathbf{P}}{\partial \mathbf{x}}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \frac{\partial^2 \mathbf{P}}{\partial \mathbf{x}\,\partial \mathbf{x}}\Delta\mathbf{x} + \cdots = \mathbf{P}_g + \delta\mathbf{P}_g + \frac{1}{2}\delta^2\mathbf{P}_g + \cdots \tag{8.18}$$

where

$\Delta\mathbf{P}$ = contribution to the overall load vector due to the design change $\Delta\mathbf{x}$.

All above contributions $\Delta\mathbf{K}$, $\Delta\mathbf{u}$ and $\Delta\mathbf{P}$ are represented by first and second order terms of the Taylor series expansion. Of course, all partial derivatives have to be taken at the current design point respresented by $\mathbf{x}^{(g)}$. Note that the first partial derivative $\frac{\partial(\ldots)}{\partial \mathbf{x}}$ is defined as a row vector such that

$$\frac{\partial(\ldots)}{\partial \mathbf{x}} = \left\{ \frac{\partial(\ldots)}{\partial x_1}, \frac{\partial(\ldots)}{\partial x_2}, \frac{\partial(\ldots)}{\partial x_3}, \ldots, \frac{\partial(\ldots)}{\partial x_n} \right\} \tag{8.19}$$

where the abbreviation (...) substitutes for the corresponding matrices and vectors, respectively. The vector $\Delta\mathbf{x}$ denotes a column vector. Thus its transposed vector can be written as

$$\Delta\mathbf{x}^T = \left\{ \Delta x_1, \Delta x_2, \Delta x_3, \cdots, \Delta x_n \right\} \tag{8.20}$$

The second partial derivative $\frac{\partial^2(\ldots)}{\partial \mathbf{x}\,\partial \mathbf{x}}$ can be understood as the Hesse matrix applied to the general term (...):

$$\frac{\partial^2(\ldots)}{\partial \mathbf{x}\,\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial^2(\ldots)}{\partial x_1 \partial x_1} & \frac{\partial^2(\ldots)}{\partial x_1 \partial x_2} & \cdot & \frac{\partial^2(\ldots)}{\partial x_1 \partial x_n} \\ \frac{\partial^2(\ldots)}{\partial x_2 \partial x_1} & \frac{\partial^2(\ldots)}{\partial x_2 \partial x_2} & \cdot & \frac{\partial^2(\ldots)}{\partial x_2 \partial x_n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{\partial^2(\ldots)}{\partial x_n \partial x_1} & \frac{\partial^2(\ldots)}{\partial x_n \partial x_2} & \cdot & \frac{\partial^2(\ldots)}{\partial x_n \partial x_n} \end{bmatrix} \tag{8.21}$$

By way of these definitions the first and second variations $\delta(\ldots)$ and $\delta^2(\ldots)$ can be analytically calculated in a similar fashion as we establish catalogs for individual finite elements.

Depending on the category of the structural optimization problem several simplifications are possible. In theory, the structural optimization could be applied to optimize all potential types of structural parameters, like geometrical optimization variables, structural entities, dimensions etc.. In practice, however, such a procedure is avoided due to the computational

effort needed to solve large-order nonlinear optimization problems. Additionally, it is not necessary to include all potential optimization variables because , due to his experience, the structural engineer is easily capable of pre-assigning a lot of eventually possible variables, and reducing the dimension n of a given problem. Rather, the designer usually modifies the structure to be optimized, locally, whereby the performance of the structural optimization can be substantially improved. Therefore, a major number of the terms of the power series can be zeroed. Furthermore, a first order Taylor expansion to linearize the approximation may be permitted in many cases. On the contrary, second order Taylor expansion may be mandatory if more complex and highly nonlinear problems occur.

In order to find a general formulation for the re-analysis problem, we only have to consider the overall system equations, and the overall contributions $\Delta\mathbf{K}$, $\Delta\mathbf{u}$ and $\Delta\mathbf{P}$ defined at the global level of the structural system. Instead of solving the stiffness problem (8.15) created due to current design changes $\Delta\mathbf{x}$

$$\hat{\mathbf{K}}\,\hat{\mathbf{u}} = \hat{\mathbf{P}}$$

we take advantage of the information derived from the previous designs. Therefore, we replace the matrix $\mathbf{K}$ by $\mathbf{K}_g+\Delta\mathbf{K}$ and the vectors $\hat{\mathbf{u}}$ as well as $\mathbf{P}$ by $\mathbf{u}_g+\Delta\mathbf{u}$ and $\mathbf{P}_g+\Delta\mathbf{P}$, respectively. Hence, we obtain

$$(\mathbf{K}_g+\Delta\mathbf{K})\;(\mathbf{u}_g+\Delta\mathbf{u}) = (\mathbf{P}_g+\Delta\mathbf{P}) \tag{8.22}$$

Multiplying and re-arranging yields

$$\mathbf{K}_g\,\Delta\mathbf{u} = \Delta\mathbf{P}-\Delta\mathbf{K}\;(\mathbf{u}_g+\Delta\mathbf{u}) \tag{8.23}$$

In the above equation the matrix $\mathbf{K}_g$ and the vector $\mathbf{u}_g$ are known from the previous analysis step, whereas the contributions $\Delta\mathbf{P}$ and $\Delta\mathbf{K}$ are approximated through first and, conditionally, second order terms of the Taylor series expansion such that

$$\Delta\mathbf{K}\approx\frac{\partial\mathbf{K}}{\partial\mathbf{x}}\Delta\mathbf{x}+\frac{1}{2}\Delta\mathbf{x}^T\frac{\partial^2\mathbf{K}}{\partial\mathbf{x}\,\partial\mathbf{x}}\Delta\mathbf{x} \tag{8.23a}$$

and

$$\Delta\mathbf{P}\approx\frac{\partial\mathbf{P}}{\partial\mathbf{x}}\Delta\mathbf{x}+\frac{1}{2}\Delta\mathbf{x}^T\frac{\partial^2\mathbf{P}}{\partial\mathbf{x}\,\partial\mathbf{x}}\Delta\mathbf{x} \tag{8.23b}$$

Again, note that the derivatives have to be evaluated at the current design point $\mathbf{x}^{(g)}$. The conceivably simpliest instance would be that one in which

(a) a linearization for the matrix $\Delta\mathbf{K}$ is permitted

(b) no load variation $\Delta\mathbf{P}$ due to geometrical variation of the structural system occurs.

In this very special case, we obtain

$$\Delta\mathbf{K} = \frac{\partial\mathbf{K}}{\partial\mathbf{x}}\Delta\mathbf{x}\;;\;\Delta\mathbf{P} = 0 \tag{8.23c}$$

It should be emphasized that with the aid of power series expansion the original implicit formulation of the structural response has been transferred towards an explicit one. Therefore, the structural response can be written as an explicit function of the optimization variables, and , of course, the structural contraints, which directly depend on the response vector, can be written in an explicit fashion. As a result, we have finally attained a form for constraints which are assumed to be standard or, at least, favored in nonlinear mathematical programming problems. (see also comments after equation (5.6)).

In equation (8.23), the only unknown quantity is the vector $\Delta\mathbf{u}$ which represents the variation of the the response $\mathbf{u}_g$ with respect to the change $\Delta\mathbf{x}$. Since the unknown vector $\Delta\mathbf{u}$ appears on the left and right hand side of equation (8.23), we have to write this equation as an iteration statement if we want to find a solution for $\Delta\mathbf{u}$:

$$\mathbf{K}_g\,\Delta\mathbf{u}^{(\omega)} = \Delta\mathbf{P}-\Delta\mathbf{K}\;(\mathbf{u}_g+\Delta\mathbf{u}^{(\omega-1)}) \tag{8.24}$$

where

(ω) = iteration counter of the iteration for the re-analysis problem.

Through this formulation, we are capable of performing the finite element analysis at any updated design without the need to resolve the complete updated stiffness equation. The term $\Delta \mathbf{u}^{(\omega-1)}$ is known from the previous iteration step. Initially, we make the guess that

$$\Delta \mathbf{u}^{(0)} = 0 \tag{8.24a}$$

Therefore, the right handside of the equation (8.24) is always known and can be understood as a pseudo-load vector. Hence,

$$\mathbf{K}_g \, \Delta \mathbf{u}^{(\omega)} = \Delta \mathbf{P}^{(\omega)} \tag{8.25}$$

with

$$\Delta \mathbf{P}^{(\omega)} = \Delta \mathbf{P} - \Delta \mathbf{K} (\mathbf{u}_g - \Delta \mathbf{u}^{(\omega-1)}) \tag{8.25a}$$

where

$\Delta \mathbf{P}^{(\omega)}$ = pseudo-load vector due to the design change $\Delta \mathbf{x}$

The iteration statement (8.25) represents linear stiffness equations due to the current design change $\Delta \mathbf{x}$. We have to re-iterate until succesive $\Delta \mathbf{u}^{(\omega)}$ values are identical with respect to a given tolerance. All popular equation solvers can be applied. However, we have to ensure that the solution is efficiently carried out because its numerical affort makes up the majority of the total effort needed within the individual iteration steps (g) of the optimization process.

An appropriate solution technique is the well known skyline or active column solution method developed by WILSON [18]. The method is based on the Gauss elimination approach but particularly takes advantage of the sparsity and the special nature of matrices involved in structural analysis problems. The first part of the solution is a factorization of the global stiffness matrix, in our case of the matrix $\mathbf{K}_g$, such that

$$\mathbf{K}_g \, \Delta \mathbf{u}^{(\omega)} = \mathbf{L}_g \, \mathbf{D}_g \, \mathbf{L}_g^T \Delta \mathbf{u}^{(\omega)} = \Delta \mathbf{P}^{(\omega)} \tag{8.26}$$

Since the stiffness matrix $\mathbf{K}_g$ is not affected by design changes, only the pseudo-load vector $\Delta \mathbf{P}^{(\omega)}$ has to be modified. It should be mentioned that the matrix $\Delta \mathbf{K}$ needs never be wholly formed since this matrix consist of a number of isolated submatrices which can be directly used to calculate the pseudo-load vector $\Delta \mathbf{P}^{(\omega)}$. This property of $\Delta \mathbf{K}$ permits the automatic creation of a valuable cut-off operation. Hence, only the backsubstitution, which is the second phase of the overall solution, must be performed, and the factorization has not to be repeated. Furthermore, the backsubstitution can be started at that portion of the linear equation system on and after which the first change of the pseudo-load vector occurs (so-called pertubation point). The advantage of being able to start from the pertubation point upwards represents a further significant cut-off operation and reduces the re-solution time, substantially. If the changes of the displacements are once calculated, the approximated displacements $\hat{\mathbf{u}}$ of the updated system, which are needed in structural constraints, can be evaluated by simple summation

$$\hat{\mathbf{u}}^{(\omega)} = \mathbf{u}_g + \Delta \mathbf{u}^{(\omega)} \tag{8.27}$$

For the purpose of a more general discussion we additionally establish a general matrix relationship for the structural respones $\hat{\mathbf{u}}$ in which all matrices involved can be identified at one glance. Substitution of the term $\Delta \mathbf{u} = \hat{\mathbf{u}} - \mathbf{u}_g$ into equation (8.23) yields

$$\mathbf{K}_g \, \hat{\mathbf{u}} = \mathbf{K}_g \, \mathbf{u}_g + \Delta \mathbf{P} - \Delta \mathbf{K} \, \hat{\mathbf{u}} \tag{8.28}$$

or

$$\hat{\mathbf{u}} = \mathbf{u}_g + \mathbf{K}_g^{-1} \, (\Delta \mathbf{P} - \Delta \mathbf{K} \, \hat{\mathbf{u}}) \tag{8.28a}$$

For the special case , $\Delta \mathbf{P} = 0$, we obtain

$$\hat{\mathbf{u}} = - \, \mathbf{K}_g^{-1} \, \Delta \mathbf{K} \, \hat{\mathbf{u}} \tag{8.28b}$$

It should be mentioned that the above relationships are not appropriate for the iterative re-analysis approach because the matrix inversion and the matrix multiplication incorporated are less efficient in computer programs than the elimination process. On the other hand, they are useful for expressing the interdependencies between the original or current design $\mathbf{u}_g$, the response of the updated design $\hat{\mathbf{u}}$ and active mechanical degrees of freedom associated with particular optimization variables, in an abstract manner. We will be taking advantage of this property when we have to discuss the approximation concept applied at the element level of the structural system. Active degrees of freedom are mechanical unknowns which are directly affected by optimization variables. To give an example, if node J of a structural system is associated with the unknown displacements u_j, v_j, w_j and if the co-ordinates of node J are varied,the node displacements are directly affected by this change and, therefore, active. Active degrees of freedom can facilitate the insight into the structural optimization problem because the structural equations can be written in a more condensed form.

After a specified number of structural approximations of the finite element equations the approximation is replaced by a complete re-analysis. This complete analysis is to compensate eventually unfeasible approximation errors which might be encountered through continuos linearizations and quadratures used within the power series expansion. Consequently, a complete re-analysis is performed after $\Delta\bar{g}$ re-analysis approximations.(re-fresh cycle). At the stage $\mathbf{x}_{g+\Delta\bar{g}}$, the complete stiffness equations can be written in the form

$$\mathbf{K}(\mathbf{x}_{g+\Delta\bar{g}}) \bullet \mathbf{u}(\mathbf{x}_{g+\Delta\bar{g}}) = \mathbf{P}(\mathbf{x}_{g+\Delta\bar{g}}) \tag{8.29}$$

or shorter

$$\mathbf{K}_{g+\Delta\bar{g}}\, \mathbf{u}_{g+\Delta\bar{g}} = \mathbf{P}_{g+\Delta\bar{g}} \tag{8.30}$$

For solution, two distinct solution methods are acceptable. Again, the skyline or active column solver discussed previously is appropriate. However, since the global stiffness matrix $\mathbf{K}_{g+\Delta\bar{g}}$ now also contains contributions due to design changes, the factorization has to be repeated starting with the first row in which a modification of the matrix $\mathbf{K}_g$ occurs.

A further method most recently propagated by NOUR-OMID and SIMON [23] is the preconditioning method. For the time being, it is unsettled which of both methods would be more advantageous for which types of optimization problems. The preconditioning algorithm links the well known conjugate gradient method with a special partitioning of the current global stiffness matrix. According to the logic of conjugate gradient methods, the residual norm of the equation system is minimized. The partitioning of the global stiffness matrix $\mathbf{K}_{g+\Delta\bar{g}}$ is performed in such a manner that

$$\mathbf{K}_{g+\Delta\bar{g}} = \mathbf{K}^{d}_{g+\Delta\bar{g}} + \mathbf{K}^{r}_{g+\Delta\bar{g}} \tag{8.31}$$

where

$\mathbf{K}^{d}_{g+\Delta\bar{g}}$ = dense part of the original matrix whose elements are concentrated within an appropriate band around the diagonal

$\mathbf{K}^{r}_{g+\Delta\bar{g}}$ = remainder of the original matrix whose elements contain the contributions outside the band of the dense part; (so-called fill-ins).

Then, the conjugate gradient process is applied to the equation system

$$\mathbf{K}^{d}_{g+\Delta\bar{g}}\, \mathbf{u}^{d}_{g+\Delta\bar{g}} = \mathbf{P}_{g+\Delta\bar{g}} \tag{8.32}$$

whereas the residual norm is calculated from the original system (8.30). Using the previous solution $\mathbf{u}_g$ as an initial guess leads to a rapid convergence towards the solution $\mathbf{u}_{g+\Delta\bar{g}}$.

8.3.2. Eigenvalue problems

As previously outlined, in the case of structural optimization of linear discrete systems the solution of eigenvalue problems can be often avoided. Since this type of structural system frequently permits the application of conventional formulas of classic structural analysis a great deal of numerical effort can be saved. Two simple examples are to be mentioned. If trusses are subjected to static loading the critical load needed for admissible bounds of the structural constraints can be easily derived from Euler's buckling load. Similarly, the critical load for frames can be often approximately calculated by using appropriate substitute-systems. In both cases no solution of an eigenvalue problem is necessary. On the other hand, of course, we have to accept a certain loss of general applicability. That is to say that the structural constraints based on such an elementary approach are considerably dependent on the structural system to be optimized. Hence, if the topology of the structural system changes, a new formulation of the constraints may be required.

Although we have assumed , up till now, that generalized eigenvalue problems of the form (5.3) and (5.4) can be replaced by more elementary relationships, for completeness, the link between structural optimization and eigenvalue problems involved is to be discussed, in general. Similarly to the re-analysis of conventional finite element equations, the crux of the matter is that all quantities involved in the eigenvalue problem become nonlinear and explicit functions of the optimization variables. Also, the eigenvalues must be repeatedly calculated during the optimization process. Therefore we have to employ methods equivalent to those applied to the re-analysis of the finite element equations (8.12) until (8.15).

The design vector $\mathbf{x}^{(g)}$ again denotes the current design stage. Then, the generalized eigenvalue problem associated with this vector can be written in the form

$$\mathbf{K}(\mathbf{x}=\mathbf{x}^{(g)}) \bullet \mathbf{v}(\mathbf{x}=\mathbf{x}^{(g)}) = \lambda(\mathbf{x}=\mathbf{x}^{(g)}) \bullet \mathbf{M}(\mathbf{x}=\mathbf{x}^{(g)}) \bullet \mathbf{v}(\mathbf{x}=\mathbf{x}^{(g)}) \tag{8.33}$$

or, according to the notation already used in (8.13), we obtain the short form

$$\mathbf{K}_g \mathbf{v}_g = \lambda_g \mathbf{M}_g \mathbf{v}_g \tag{8.34}$$

If the design vector $\mathbf{x}^{(g)}$ is modified from $\mathbf{x}^{(g)}$ to $\mathbf{x}^{(g)}+\Delta\mathbf{x}$ we have

$$\mathbf{K}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) \bullet \mathbf{v}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) = \lambda(\mathbf{x}^{(g)}+\Delta\mathbf{x})\ \mathbf{M}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) \bullet \mathbf{v}(\mathbf{x}^{(g)}+\Delta\mathbf{x}) \tag{8.35}$$

or shorter

$$\hat{\mathbf{K}}\,\hat{\mathbf{v}} = \hat{\lambda}\,\hat{\mathbf{M}}\,\hat{\mathbf{v}} \tag{8.36}$$

However, instead of solving the complete updated problem (8.36), we solve the less costly problem

$$(\mathbf{K}_g+\Delta\mathbf{K})\ (\mathbf{v}_g+\Delta\mathbf{v}) = (\lambda_g+\Delta\lambda)\ (\mathbf{M}_g+\Delta\mathbf{M})\ (\mathbf{v}_g+\Delta\mathbf{v}) \tag{8.37}$$

This problem is less costly because the assemblage is exactly restricted to only those contributions which result from design modifications. In equation (8.37), we approximate the two contributions $\Delta\mathbf{K}$ and $\Delta\mathbf{M}$, respectively, by Taylor power series expansion, analogously to the re-analysis of conventional finite element equations. Thus, along with the approximation of the matrix $\Delta\mathbf{K}$ (see equation (8.16)), we have to expand the contribution $\Delta\mathbf{M}$.

$$\Delta\mathbf{M} = \Delta\mathbf{M} \approx \mathbf{M}_g + \frac{\partial\mathbf{M}}{\partial\mathbf{x}}\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T\frac{\partial^2\mathbf{M}}{\partial\mathbf{x}\,\partial\mathbf{x}}\Delta\mathbf{x} + \cdots = \delta\mathbf{M}_g + \frac{1}{2}\delta^2\mathbf{M}_g + \cdots \tag{8.38}$$

Since the terms $\mathbf{K}_g$, $\mathbf{M}_g$ and λ_g are also known from the previous solution at the stage $\mathbf{x}=\mathbf{x}^{(g)}$, only the term $\Delta\lambda$ is unknown. (The eigenvectors $\mathbf{v}_g$ and $\Delta\mathbf{v}$ are not of interest for the evaluation of structural constraints). Then , we have to solve

$$\left[(\mathbf{K}_g+\Delta\mathbf{K})-\lambda_g\ (\mathbf{M}_g+\Delta\mathbf{M})\right]\ (\mathbf{v}_g+\Delta\mathbf{v}) = \Delta\lambda\ (\mathbf{M}_g+\Delta\mathbf{M})\ (\mathbf{v}_g+\Delta\mathbf{v}) \tag{8.39}$$

Using the abbreviations

$$\Delta\mathbf{K}^* = \left[(\mathbf{K}_g+\Delta\mathbf{K})-\lambda_g\ (\mathbf{M}_g+\Delta\mathbf{M})\right] \tag{8.40}$$

$$\Delta \mathbf{v}^* = (\mathbf{v}_g + \Delta \mathbf{v}) \tag{8.41}$$

and

$$\Delta \mathbf{M}^* = (\mathbf{M}_g + \Delta \mathbf{M}) \tag{8.42}$$

we can establish a modified eigenvalue problem which can be written as follows

$$\Delta \mathbf{K}^* \Delta \mathbf{v}^* = \Delta\lambda \, \Delta \mathbf{M}^* \Delta \mathbf{v}^* \tag{8.43}$$

After having calculated the eigenvalues $\Delta\lambda$ the actual eigenvalues $\hat{\lambda}$ can be found by simple summation

$$\hat{\lambda} = \lambda_g + \Delta\lambda \tag{8.44}$$

Again, to eliminate undesirable approximation errors after a certain number of approximations a complete solution of the eigenproblem ought to be carried out (re-fresh cycle).

The problems (8.34), (8.36) and (8.43) define symmetric generalized eigenvalue problems because of the nature of the matrices derived from structural analysis problems. In the majority of cases these matrices are also positive definite. If structural systems with a major number of degrees of freedom are considered, in addition, sparse matrices occur.

With respect to structural optimization problems we are only interested in the lowest or , if dynamical loadings might be encountered, in the first lowest eigenvalues. For this special category of problems three solution algorithms are acceptable. If we want to calculate the lowest eigenvalue (e.g. if the lowest critical buckling load is sought) an appropriate method is

(a) the vector iteration sufficiently known [24].

If we have to calculate more eigenvalues than the lowest one

(b) the subspace iteration technique [24]

(c) the Lanczos algorithm [25], [26]

are preferrable. In particular the last method has recently become very attractive because this method takes advantage of the sparsity of the involved matrices. Additionally, its speed is reported to be more than one order of magnitude faster than other popular competitors. Both properties are exceptionally significant in the context of structural optimization problems, for which an efficient structural analysis is always very crucial.

8.4. Relationships at element and optimization-variable level

Global matrices defined at the overall system level can be separately evaluated either at the element level or at the level of individual optimization variables. The single contributions can be calculated at each of both levels and, then, successively assembled into the corresponding matrix defined at the global system level. That means that the assemblage pattern applied to structural optimization models is vastly analogous to that of finite element models, provided that linear elastic structures are considered.

Within the scope of this report, and with regard to the comments made in the preceding paragraph, we will be exclusively dealing with equilibrium field problems. Subsequently, eigenvalue problems are suspended.

Referring to equations (8.24) and (8.28) we have to establish the two matrix-contributions $\Delta\mathbf{P}$ and $\Delta\mathbf{K}$. Neglecting dead loads we can assume that the term $\Delta\mathbf{P}$ vanishes, and we have only to consider those modifications represented by the matrix $\Delta\mathbf{K}$. It should be pointed out that, by no means, is such a simplification purely academic. In practice, loaded nodes and the loading to which these nodes are subjected are fixed in the majority of instances, and so the simplification made is reasonable. (Within the general optimization of continuous structures, of course, a more complex situation arises due to the more comprehensive discretization process).

With regard to vanishing $\Delta\mathbf{P}$ - terms, the assemblage of the pseudo-load vector is primarily dominated by the assemblage of the global matrix $\Delta\mathbf{K}$. Let us start with the element related approach. We can form the matrix $\Delta\mathbf{K}$ by addition of element related terms, element by element, so that

$$\Delta\mathbf{K} = \sum_{e=1}^{e=N} \Delta\mathbf{K}_e = \sum_{e=1}^{e=N} \Delta\mathbf{K}_e(\mathbf{x}_e) \tag{8.45}$$

where

e current element number ; e=1,2,3,...,N

N total number of elements (or members, respectively)

$\Delta\mathbf{K}_e$ contribution of the element e associated with optimization variables concentrated in the vector $\mathbf{x}_e$

$\mathbf{x}_e$ optimization vector specificly introduced for the element e ; $\mathbf{x}_e \subset \mathbf{x}$; $\mathbf{x}_e, \mathbf{x} \in S^n$.

Following the second line, we can assemble the $\Delta\mathbf{K}$ - matrix from the base of the isolated optimization variable by utilizing the subscript i instead of e. While the first approach permits us to follow the organisation of the finite element model, the second is optimization-model-oriented. However, both approaches must yield the same result.

Using the individual optimization variable x_i of the general design vector $\mathbf{x}$ leads to the following assemblage statement:

$$\Delta\mathbf{K} = \sum_{i=1}^{i=n} \Delta\mathbf{K}_i = \sum_{i=1}^{i=n} \Delta\mathbf{K}_i(x_i) \tag{8.46}$$

where

$\Delta\mathbf{K}_i$ contribution to the global (stiffness) matrix $\Delta\mathbf{K}$ due to the optimization variable x_i.

The second approach was at first proposed by WANG and PILKEY [27], and can be regarded as an efficient parameterization method for re-analysis formulations. Equating the two equations (8.45) and (8.46) yields

$$\sum_{i=1}^{i=n} \Delta\mathbf{K}_i = \sum_{e=1}^{e=N} \Delta\mathbf{K}_e = \Delta\mathbf{K} \tag{8.47}$$

which permits mapping of the two distinct groups of submatrices $\Delta\mathbf{K}_i$ and $\Delta\mathbf{K}_e$. Such a mapping may be important for computer program realisations if one wishes to make use of a given finite element data base. The mapping is represented by the following assignment statement:

$$\sum_{i=1}^{i=n} \sum_{e=a_i}^{e=b_i} \Delta\mathbf{K}_e = \sum_{e=1}^{e=n} \sum_{i=a_e}^{i=b_e} \Delta\mathbf{K}_i = \Delta\mathbf{K} \tag{8.47a}$$

where the subscript-pairs of the second sum (a_i , b_i) or (a_e , b_e) , respectively, are the initial and limit values depending on the current subscripts of the first sum i or e, respectively.

We proceed by condensing the system of equations already described in equation (8.28) which , for convenience, may be once more repeated. We have

$$\hat{\mathbf{u}} = \mathbf{u}_g + \mathbf{K}_g^{-1} (\Delta\mathbf{P} - \Delta\mathbf{K}\,\hat{\mathbf{u}}) = \mathbf{u}_g - \mathbf{K}_g^{-1}\,\Delta\mathbf{K}\,\hat{\mathbf{u}} \tag{8.48}$$

because of the assumption that $\Delta\mathbf{P} = 0$.

In order to be capable of making use of the parametrization approach we now define

$\mathbf{u}_i$ structural response vector of the active degrees of freedom associated with optimization variable x_i

and

$N_{act,i}$ number of the active degrees of freedom associated with optimization variable x_i.

where the vector $\mathbf{u}_i$ contains $N_{act,i}$ components. Also, we define

$\overline{\Delta \mathbf{K}}_i$ condensed contribution matrix of the $\Delta \mathbf{K}_i$ - matrix associated with the optimization variable x_i, having the order $O(N_{act,i} \times N_{act,i})$.

The condensed matrix $\overline{\Delta \mathbf{K}}_i$ is identical to the original matrix $\Delta \mathbf{K}_i$; however, all rows and columns containing zeros are deleted. This property can be abstractly described by the Boolean matrix $\mathbf{B}_i$, where the positions of the values 0 and 1 create the desired condensation effect so that

$$\Delta \mathbf{K}_i = \mathbf{B}_i^T \overline{\Delta \mathbf{K}}_i \mathbf{B}_i \tag{8.49}$$

Note that, in practice, the Boolean matrix only involves the selection of the proper rows and columns as when assembling conventional global stiffness matrices. That means that products involving Boolean matrices need not be carried out.

Substituting $\Delta \mathbf{K}$, according to equation (8.40), and $\Delta \mathbf{K}_i$ into equation (8.48) leads to

$$\hat{\mathbf{u}} = \mathbf{u}_g - \mathbf{K}_g^{-1} \left(\sum_{i=1}^{i=n} \Delta \mathbf{K}_i \right) \hat{\mathbf{u}} \tag{8.50}$$

or

$$\hat{\mathbf{u}} = \mathbf{u}_g - \mathbf{K}_g^{-1} \sum_{i=1}^{i=n} \left(\mathbf{B}_i^T \overline{\Delta \mathbf{K}}_i \mathbf{B}_i \hat{\mathbf{u}} \right) \tag{8.51}$$

We can also write

$$\hat{\mathbf{u}} = \mathbf{u}_g - \mathbf{K}_g^{-1} \sum_{i=1}^{i=n} \left(\mathbf{B}_i^T \overline{\Delta \mathbf{K}}_i \right) \left(\mathbf{B}_i \hat{\mathbf{u}} \right) \tag{8.52}$$

where the symbol $\sum$ acts on both expressions enclosed in parentheses. Note that

$$\mathbf{u}_i = \mathbf{B}_i \hat{\mathbf{u}} \tag{8.53}$$

Therefore, we obtain

$$\hat{\mathbf{u}} = \mathbf{u}_g - \mathbf{K}_g^{-1} \sum_{i=1}^{i=n} \left(\mathbf{B}_i^T \overline{\Delta \mathbf{K}}_i \mathbf{u}_i \right) \tag{8.54}$$

Furthermore, the particulars $\overline{\Delta \mathbf{K}}_i$ can be expanded as Taylor power series, correspondingly, to equation (8.13). Due to the chain rule, we have

$$\Delta \mathbf{K}_i = \Delta \left(\mathbf{B}_i^T \overline{\mathbf{K}}_i \mathbf{B}_i \right) = \Delta \mathbf{B}_i^T \overline{\mathbf{K}}_i \mathbf{B}_i + \mathbf{B}_i^T \Delta \overline{\mathbf{K}}_i \mathbf{B}_i + \mathbf{B}_i^T \overline{\mathbf{K}}_i \Delta \mathbf{B}_i \tag{8.55}$$

However, since the derivatives of Boolean matrices must vanish

$$\Delta \mathbf{B}_i^T = \Delta \mathbf{B}_i = 0 \tag{8.56}$$

we know that

$$\Delta \mathbf{K}_i = \mathbf{B}_i^T \overline{\Delta \mathbf{K}}_i \mathbf{B}_i \tag{8.57}$$

Since the matrices $\Delta \mathbf{K}_i$ only depend on individual optimization variables, the Taylor power series expansion can be simplified by the following expression

$$\Delta \mathbf{K}_i = \frac{d\mathbf{K}_i}{dx_i} \Delta x_i + \tfrac{1}{2} \frac{d^2 \mathbf{K}_i}{dx_i^2} \Delta x_i^2 \tag{8.58}$$

where the derivatives are to be taken at the current design stage $\mathbf{x}^{(g)}$. Therefore, the relationship

$$\Delta \mathbf{K}_i = \mathbf{B}_i^T \left(\frac{d\overline{\mathbf{K}}_i}{dx_i} \Delta x_i \right) \mathbf{B}_i + \tfrac{1}{2} \mathbf{B}_i^T \left(\frac{d^2 \overline{\mathbf{K}}_i}{dx_i^2} \Delta x_i^2 \right) \mathbf{B}_i \tag{8.59}$$

must be valid. Now, after re-arranging, we can write

$$\hat{\mathbf{u}} = \mathbf{u}_g - \mathbf{K}_g^{-1} \sum_{i=1}^{i=n} \mathbf{B}_i^T \left[\frac{d\overline{\mathbf{K}}_i}{dx_i} \Delta x_i + \tfrac{1}{2} \frac{d^2 \overline{\mathbf{K}}_i}{dx_i^2} \Delta x_i^2 \right] \mathbf{u}_i \tag{8.60}$$

Equation (8.60) is a relationship between the modified system response $\hat{\mathbf{u}}$, the original response $\mathbf{u}_g$, the active degrees of freedom associated with the indivudual optimization variables x_i, and these variables themselves. Obviously, all main entities and quantities of the distinct levels of both the structural and the optimization model are incorporated in the same equation. Therefore, the above equation is specified by a particular name, and is designated as **" interface equation of the structural optimization model ".**

9. Conclusion

The present report forms the conceptional and abstract base for a comprehensive solution of structural optimization problems associated with multiple finite element analysis. It may contribute to a better understanding and wider use of coupled problems in the new field of numerical optimization and repetitive finite element applications. The computer implementation of the elaborated concept is currently under investigation and appears very promising.

10. Outlook on future expansions and improvements

A final note should be made on potential improvements and expansions which are to be accomplished in the nearest future. With regard to evolution strategies the following aims will be pursued:

(a) development of an interactive computer aided and computer graphic design tool to provide better information for controlling the optimization process

(b) automatic elimination of active constraints by means of automaticly augmented optimization criteria (expressed in the terminolgy of the evolution theory : changing quality of environment)

(c) enbedding of additional optimization mechanismi as already suggested

(d) creation of vectorizable codes in order to take advantage of the parallel organisation of the (μ,λ) - strategy.

As far as the structural analysis line is concerned, the following issues will be investigated:

(e) comparison of the efficiency of acceptable equation solvers with respect to the particular requirements of structural optimization problems

(f) implementation of the numerical solution of general eigenproblems to automatize the establishment of structural constraints.

11. References

[1] R.W. CLOUGH, *The finite element method in plane stress analysis.* Proceedings of the 2nd Conference on Electronic Computations, ASCE, 1960

[2] L.A. SCHMITT, *Structural design by systematic systhesis.* Proceedings of the 2nd Conference on Electronic Computations, ASCE, 1960

[3] A.HOEFLER, *Shape optimization of light-weight trusses by means of the evolution strategy,* Dissertation, TU-Berlin, D 83, (in German), 1976

[4] D. HARTMANN, *A generally applicable structural optimization method and its application to cylindrical shells,* Engineering Optimization, Vol.3, (1978), pp. 201 -213

[5] D. HARTMANN, *Optimization of hyperbolic paraboloidal shells - A problem of algorithmically nonlinear programming,* 20.IASS-World Congress, Madrid, September 1979, pp. 5.265 - 5.277

[6] D. HARTMANN, *Optimization in CAD - On the applicability of nonlinear evolution strategies for optimization strategies in CAD,* IFIP-Conference on Optimization in CAD, Lyon, France, 1983

[7] D.HARTMANN, *Computer aided-structural optimization - Enbedding of the finite element method into numerical nonlinear optimization,* ASME, Design Automation Conference, September 11 -14, Detroit, Michigan, USA, Paper No. 83 - DET - 59, 1983

[8] W.C. CARPENTER ; E.A. SMITH, *Computational efficency in structural optimization,* Engineering Optimization, Vol. 1, (1975), pp. 169 - 188

[9] S.L. LIPSON ; L.B. GWIN, *The complex method applied to optimal truss configurations,* Comp. Struct., Vol. 7, (1977), pp. 469 - 475,

[10] H.P. SCHWEFEL, *Numerical optimization of computer models,* John Wiley & Sons , New York , 1982

[11] R. FLETCHER, *Conjugate gradient methods,* in, *Numerical methods for unconstrained optimization,* Academic Press, London, 1972

[12] M. LAWO, *Optimum design of framed structures subjected to dynamic loads,* Dissertation, University of Essen, Germany, (in German), 1981

[13] G. HEUSENER, *Optimization of nuclear power systems,* Nuclear Research Center, Karlsruhe, Germany, Report No. KFK 1238, 1970

[14] R.R. CEMBROWICZ ; G.E. KRAUTER, *Optimization of urban and regional water supply systems,* Proceedings on the Conference on System Approach for Development, Cairo, Egypt, 1977

[15] I. RECHENBERG, *Evolution Strategy : Optimization of technical systems by virtue of principles of biological evolution.* Fromann-Holzboog Verlag, Stuttgart,(in German), 1977

[16] D. HARTMANN, *Computer Aided Structural Optimization by means of Evolution Strategies.* Report UCB/SESM-84/7, University of California, Berkeley, Structural Engineering and Structural Mechanics, June 1984

[17] E.L. WILSON; K.J. BATHE; W.P. DOHERTY, *Direct solution of large systems of linear equations.* Computer and Structures, 1974, vol.4, 363

[18] E.L. WILSON; H.H. DOVEY, *Solution or reduction of equilibrium equations for large structural systems.* Advances in Engineering Software, 1978, Vol.1 , No.1

[19] J.S. ARORA, *Survey of Structural Reanalysis Techniques.* ASCE, Journal of the Structural Division, Vol. 102 , No. ST4, Apr. 1976, pp. 783 - 802

[20] U. KIRSCH, *Approximate Structural Reanalysis Based on Series Expansion.* Computer Methods in Applied Mechanics and Engineering, Vol. 26, No.2 , pp. 205 - 223, 1981

[21] J.E. WHITESELL, *Power Series for Real Symmetric Eigensystems.* Report UM-MEAM-82-6, Department of Mechannical Engineering adn Applied Mechanics, The University of Michigan , Ann Arbor, 1982

[22] J.E. WHITESELL, *Rational Approximants in Structural Design Reanalysis.* ASME, 83-Det-70, Design and Production Engineering Conference, Dearborn, Michigan, Sept. 11 -14 , 1983

[23] N.O. BAHRAM; H.D. SIMON, *Banded Preconditioning For The Solution Of Symmetric Positive Definite Linear Systems.* Report of Center For Pure and Applied Mathematics, University of California, Berkeley, PAM-185, 1983

[24] K.J. BATHE; E.L. WILSON, *Solution methods for eigenproblems in structural mechanics.* Int. J. Num. Meth. Engng. , vol. 6 , pp. 213 -226, 1976

[25] B.N. PARLETT, *The Symmetric Eigenvalue Problem.* Prentice Hall, Englewood, 1980

[26] D.S. SCOTT, *The Lanczos algorithm.*
in:
I.S. DUFF, *Sparse Matrices and their Uses.* Academic Press , London, New York, 1981

[27] B.P. WANG ; W.D. PILKEY, *Efficient Reanalysis of modified structures.* Proceedings of the First Chautugua on Finite Element Modeling, Schaefer Analysis, 1980