Structural Optimization 6, 233–237 (1993)

Structural Optimization

Automatic evaluation of move-limits in structural optimization

G.M. Fadel

Department of Mechanical Engineering, Clemson University, Box 340921, Clemson, SC 29634-0921, USA

S. Cimtalay

School of Mechanical Engineering, Georgia Tech, USA

Abstract The two-point exponential approximation method was introduced by Fadel *et al.* (1990) and tested on structural optimization problems with stress and displacement constraints. It was subsequently tested on problems with frequency constraints (Sareen *et al.* 1991). The results reported in earlier papers showed reductions in the number of iterations needed to reach an optimum, and a smoother convergence towards that optimum. The method, which consists in correcting Taylor series approximations using previous design history, is used in the present paper to automatically determine move-limits. Move-limits are the allowable changes in design variables during the optimization of the approximate problem. The exponents, computed in the two-point exponential approximation by matching slopes between two design iterations, are used as a measure of non-linearity of the objective function and constraints with respect to each of the design variables. The relationships between the move-limits and the exponents are established and individual move-limits are computed and applied to each design variable. The method is applied to two classical structural optimization examples. It provides the engineer with more flexibility when choosing the move-limits and typically converges in less iterations.

1 Introduction

In the practice of optimization, especially when complex structural, thermal, aerodynamic or other analyses are needed, the computer time required to perform the analyses is critical. Most large optimization problems have been formulated such that the number of full scale analyses are minimal. This is generally accomplished by reducing the original problem to an approximate, simpler model which can be optimized within certain additional constraints. A full scale analysis of the original problem is used to obtain initial results and the sensitivity of the objective and constraints to the design variables. Using this information, an approximate problem is formulated, then optimized. The original problem is then solved again with the design variables obtained from the optimized approximate problem and the procedure is reiterated until overall convergence is attained. The critical aspect of the procedure is the quality of the approximation. For a very highly non-linear problem, linear approximations are valid only in a very small domain around the original design point, whereas in better behaved problems, larger moves can be accomplished. The trade-off between the quality of approximations and the number of full scale analyses is what dictates the overall time needed to reach the optimum (if at all reachable). The accepted procedure for solving such problems consists in selecting a design state, setting up the move-limits, i.e. the acceptable relative change in the design variables where the approximations are expected to yield reasonable results, and optimizing the approximate problem. The process is then reiterated until overall convergence is obtained. Presently, engineers use their experience to decide on the magnitude of move-limits. The non-linearities of the functions involved can be assessed, and typically, uniform move-limits are imposed on all the design variables. At each iteration, the progress of the optimization is monitored, and appropriate changes in move-limits can be imposed. Occasionally, backtracking is needed and the move-limits have to be reduced.

2 Previous research

There is little evidence of previous research on the subject of automatic evaluation of move-limits. Bloebaum (1991) used expert system rules and the "effectiveness coefficient" to automatically generate individual move-limits. She demonstrated a reduction in the number of iterations needed to reach an optimum when using tailored move-limits. Her contention is stated in the following quote: "If certain design variables can be identified as having the most impact on a design and therefore requiring more restrictive move-limits, it would be possible to allow the less critical design variables more leeway in their associated move-limits." In order to assess this impact, Bloebaum used the effectiveness coefficients which represent the ratio of the slopes of the objective function and the cumulative constraint with respect to each design variable. She then evaluated the mean effectiveness coefficient, and the standard deviation of the individual coefficients from the mean was used to define maximum and minimum move-limits. Design variables with effectiveness coefficients falling between the upper and lower values were assigned move-limits based on a linear distribution between the bounds. Expert system rules were then used to either restrict or increase the move-limits based on heuristic rules involving the status of the constraints (violated, active, not active).

In this paper, we propose to use the information gained during the construction of the approximations to better understand the behaviour of the individual functions, and to automatically assess the magnitude of the move-limits.

3 Derivation of the two-point exponential approximation

Several traditional first-order approximation methods were summarized in a paper by Fadel *et al.* (1990), ranging from the simple Taylor series in the form

$$g(X) = g(X_0) + \sum (x_i - x_{i0}) \frac{\partial g}{\partial x_i},$$

to the reciprocal, hybrid, and higher-order approximations. Fadel then introduced the two-point exponential approximation which is an extension of the simpler Taylor series, adjusted by matching the derivatives at the previous design point. This correction term is incorporated into an exponent which is computed after each full analysis for each constraint, and with respect to each design variable. The exponent acts as a measure of goodness of fit. If the linear approximation is valid for a certain constraint, the exponent is equal to 1, if the reciprocal approximation is more appropriate, the exponent approaches or is equal to -1. In other cases, the exponent varies between -1 and 1, correcting, and improving the approximation. The upper and lower limits for the exponent have been imposed to control the impact of a design variable on a particular constraint. The exponents are actually computed and stored, but during the evaluation of the approximation, the appropriate limits are imposed. These limits, although conservative, have been determined by numerical experimentation (Fadel *et al.* 1990).

The two-point exponential approximation is derived as mentioned earlier by matching the slopes at previous design points. Initially, one substitutes x^{p_i} for x in the Taylor series

$$g(X^p) = g(X_0^p) + \sum (x_i^{p_i} - x_{0i}^{p_i}) \frac{\partial g}{\partial x^{p_i}},$$

and after resubstitution, one can write

$$g(X) = g(X_0) + \sum \left[\left(\frac{x_i}{x_{0i}} \right)^{p_i} - 1 \right] \left(\frac{x_{0i}}{p_i} \right) \frac{\partial g}{\partial x_i}(X_0),$$

with the exponent evaluated according to

$$p_i = 1 + \frac{\log \left[\frac{\partial g}{\partial x_i}(X_1) \right] - \log \left[\frac{\partial g}{\partial x_i}(X_0) \right]}{\log(x_{i1}) - \log(x_{x0})}.$$

The point X_1 refers to the design point at the previous iteration and X_0 refers to the current design point from where the approximation is carried out. Note that at the first iteration, since no previous design history exists, a linear or reciprocal step is carried out, depending on the preference of the user.

What can one learn from this approximation? An exponent P is computed for each function (objective and constraints) and with respect to each design variable. Essentially, a matrix of exponents is computed

$$\begin{matrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mn} \end{matrix}.$$

In this case, we have n design variables and m functions (constraints + objective). This is the same number of unknowns as the first-order derivatives. However, the additional information gathered includes some measure of curvature since the exponent in effect introduces a change of coordinate system (look also at the expression for P which is similar in form to the second-order derivative). Basically, instead of a linear approximation of the function with respect to the original variables, a linear approximation of the function with respect to the original variables raised to some exponent is built. This will not include cross terms like in the case of a second-order approximation, but it will result in a better linear approximation without the additional cost of computing the Hessian. We suggest therefore to use this information to estimate the range of validity of the new approximation.

4 Derivation of move-limits

In the process of optimizing, move-limits are imposed on the design variables. Whether these limits are uniformly imposed, or individually selected, the move-limits apply to the design variables and not to the functions (this could be done on dual problems). Some functions are, however, better behaved than others, and the move-limits have to be conservatively established so that the safest relative changes will not result in the optimizer being driven deep into the unfeasible space. Thus, most non-linear functions have to determine the magnitude of the move-limits.

Two cases have to be considered. First, if the exponents relative to one design variable (a column in the exponent matrix) are all within the range (-1 to +1), then the approximations with respect to that particular variable try to fit as accurately as possible the real functions. In such a case, the maximum move-limit should be allowed for this design variable.

Second, if one of the exponents is outside the conservative limits, then the move-limit has to be restricted in some inversely proportional way to the value of the exponent. The reason for this choice will be demonstrated below, but intuitively, the higher the exponent, the more non-linear is the function, and the tighter the move-limits should be. Also, whether the computed exponent is larger than 1 or smaller than -1, the restrictions could be different.

In order to quantify the magnitude of the move-limits, the case where the computed exponent is larger than 1 was considered first. In this case, a certain error is introduced by using a maximum exponent of +1 instead of the computed value. If $g_p(x)$ is the value obtained with the exponent p, and $g_1(x)$ is the value of the function computed with an exponent of 1, then the two approximations (considering a single variable) are

$$g_p(x) = g(x_0) + \left[\left(\frac{x}{x_0} \right)^p - 1 \right] \frac{x_0}{p} \frac{\partial g}{\partial x},$$

and

$$g_1(x) = g(x_0) + (x - x_0) \frac{\partial g}{\partial x}.$$

Assuming a relative change of A% in the design variable

$$x = x_0 + A x_0,$$

the error in the approximations can be obtained by subtracting the following two expressions:

$$\Delta g(x) = \left[\frac{(1+A)^p - 1}{p} - A \right] x_0 \frac{\partial g}{\partial x},$$

or

$$\frac{\Delta g(x)}{x_0 \frac{\partial g}{\partial x}} = W_1 = \frac{(1+A)^p - 1}{p} - A.$$

This results in a relationship between the exponent and the magnitude of the relative increase in the design variable. This relationship shows that for an acceptable error in the approximation Δg, depending on the location of the design variable (larger or smaller than 1), and on the sensitivity information (derivative of the function with respect to the design variable), one can construct a curve that illustrates the change in relative increase of the variable (A) with respect to the exponent p. This in effect is the relationship between the move-limit and the computed exponent.

Using Mathematica, the relationship between A and p for various values of the "error term" W_1 is displayed (Fig. 1.). The curves represent the decrease of the move-limit which is necessary to maintain a constant error as the calculated exponent grows from 1 to 10. The different curves represent different values of the allowable error which increases as the curves move away from the origin.

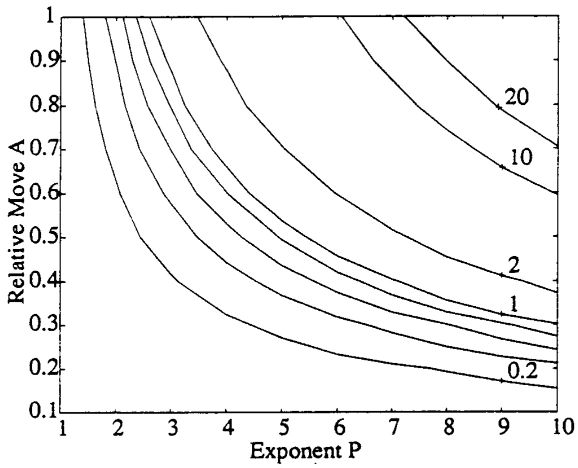

Fig. 1. Relationship between move-limit and exponent when $p > 1$

The same derivation can be performed for the negative exponent. In this case, the function g_1 is given by

$$g_1(x) = g(x_0) + (x - x_0)\frac{x_0}{x}\frac{\partial g}{\partial x},$$

and the error term becomes:

$$\Delta g(x) = \left[\frac{(1+A)^p - 1}{p} - \frac{A}{A+1}\right] x_0 \frac{\partial g}{\partial x},$$

or

$$\frac{\Delta g(x)}{x_0 \frac{\partial g}{\partial x}} = W_2 = \frac{(1+A)^p - 1}{p} - \frac{A}{A+1}.$$

The contour graph of this function is illustrated in Fig. 2, and shows that the move-limit should be decreased as the exponent decreases below -1.

As mentioned earlier, the multiplicity of curves is due to the selection of the term $\Delta g(x)/[x_0\,(\partial g/\partial x)]$. This term could be accurately computed since all the information is available, however, this would add an undue computational burden. The calculation would have to be carried out for each exponent and with respect to each design variable. The resulting moves would then have to be compared and the smallest one selected.

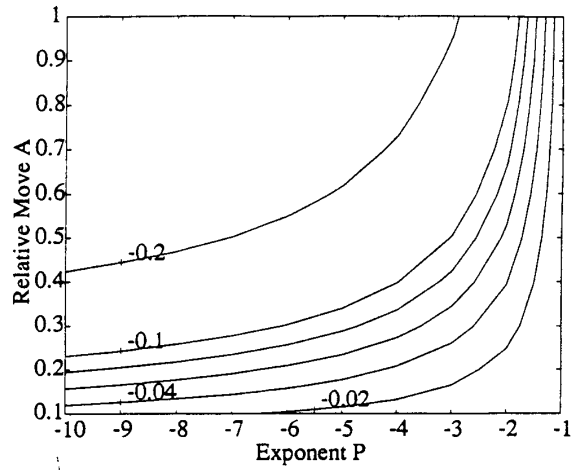

Fig. 2. Relationship between move-limit and exponent when $p < -1$

To avoid this computational burden, the method presented here approximates the behaviour of the function relating the move-limit to the exponent. The highest or smallest exponent in a column (for a specific variable) is evaluated and the move-limit is computed once for each exponent. The method consists then in selecting the move-limit for each design variable according to the function described in Fig. 3. The two sides of the "mesa" are exponentially decreasing functions which are derived by interpolation techniques from the families of contour plots illustrated in Figs. 1 and 2. Since the smallest error is desired, the lowest curve is selected for the proof of concept.

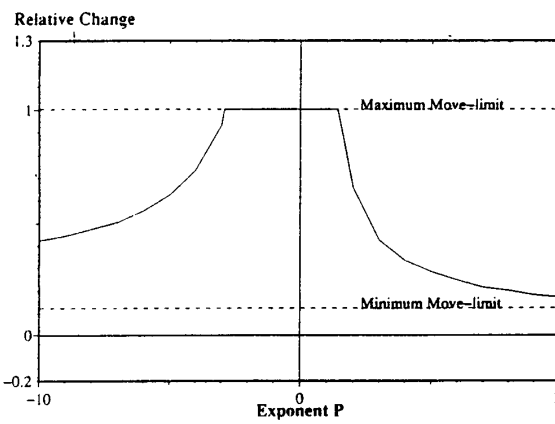

Fig. 3. "Mesa" function relating move-limits to exponents

Additionally, to provide the engineer with flexibility and control over the process, maximum and minimum move-limits are asked for once at the start of the optimization cycle. This provides the user with a certain level of control since the type of problem often dictates the non-linearity of the problem. These limits are used within the algorithm to compute individual move- limits at each iteration and for each design variable. The "mesa" illustrates relative move-limits which are bounded by the minimum and maximum move-limits provided by the user. If all the exponents fall within the +1, -1 range, then the maximum allowable move is the maximum move specified by the user. Otherwise, the magnitude of the smallest or largest exponent is used to compute the actual move which can range from the maximum to the minimum allowable limits and decreases exponentially on both sides of the range.

5 Numerical applications

Several test cases were used to test the methodology. The finite element program STAP (Bathe and Wilson 1976) was connected to the optimizer program CONMIN (Vanderplaats 1973) and used in the study. In all test cases, we compared the results with those of published literature and used a termination criterion set at 0.001.

The first test presented in this paper is the standard ten-bar truss problem with 10 design variables and 10 stress constraints (Haug and Arora 1979) (Fig. 4). The design data for the problem are $E = 10^4$ ksi, $R = 0.1$ lb/in^3, minimum cross-sectional areas = 0.1 in^2, initial value of cross-sectional areas = 10 in^2, stress limit = 25 ksi and one loading condition: -100 lbs at nodes 2 and 4 in the y direction.

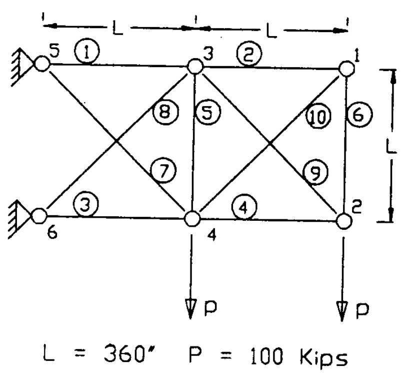

Fig. 4. Ten-bar truss example

The numbers of iterations needed for convergence are illustrated in Table 1. The variable move-limit strategy "mesa" is compared with the traditional fixed move-limit method. Several values for the maximum move-limits are tested and reported. Note that the results match published results, and that both fixed move-limits and variable move-limits converge to essentially the same values (Table 2).

Table 1. Results of the ten-bar truss example

Fixed move-limits	10%	25%	50%	75%	99%
Number of iterations	36	20	17	13	21

Variable move-limits Maxmv - Minmv	75% - 10%	99% - 10%
Number of iterations	11	14

The variable move-limit method in this case effectively reduces the number of iterations required to reach the optimum. When the fixed move-limits are very tight, the optimizer requires an excessive number of iterations to finally converge on the optimum. The question becomes: how can one select the appropriate move-limit that will result in the least amount of iterations. The variable move-limit gives the engineer more latitude in the selection of the maximum move-limit and typically converges faster. In all variable move-limit cases studied, the minimum move-limit was set at 10%. Obviously, the minimum does play a role in the number of iterations required, and research to quantify this effect is currently pursued.

The next example is a twenty five-bar transmission tower with stress and displacement constraints. The tower example is taken from the book by Haug and Arora (1979) and is illustrated in Fig. 5.

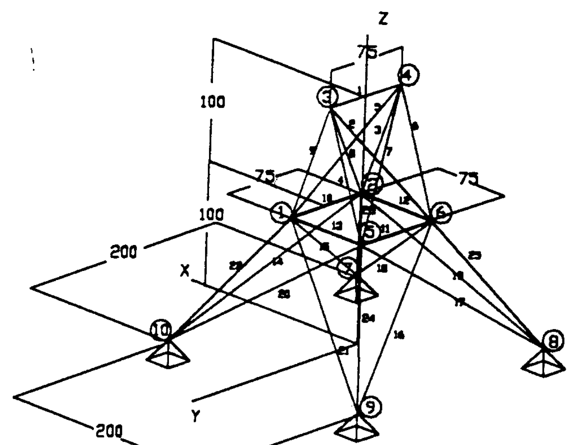

Fig. 5. 25-bar transmission tower (Haug and Arora 1979)

The problem consists of 7 design variables (due to symmetry) and 62 constraints. Fifty of these are stress constraints in compression and extension, and 12 are displacement constraints. The design data for this problem are $E = 10^4$ ksi, $R = 0.1$ lb/in^3, minimum cross-sectional areas = 0.01 in^2, initial value of cross-sectional areas = 5 in^2, stress limit = 40 ksi, displacement constraints 0.35 in all directions (nodes 3 and 4) and two loading conditions illustrated in Table 3.

The numbers of iterations needed for convergence for this case are illustrated in Table 4. Again, the results compare very closely to published data (Table 5).

The 10% fixed move-limits case converges to a value somewhat higher than the correct result, which means that we should tighten the error to converge on the correct result. The number of iterations would then be very large.

The main observations from these results are the following:

- The variable move-limit methodology contributes to the reduction in the number of iterations required to reach the optimum.
- Because of its self correcting process, the method seems to allow more leeway in selecting the move-limits. In the case of the fixed move-limits procedure, the optimizer was not able to converge in a reasonable number of iterations (less than 100) if the move-limit was set at more than 25%.

Table 2. Comparison of results for 10-bar truss problem

Member No.	Reference (in^2) area	Results (in^2) area
1	7.9379	7.9396
2	0.1	0.1
3	8.0621	8.0652
4	3.9389	3.9379
5	0.1	0.1
6	0.1	0.1
7	5.7447	5.7462
8	5.569	5.5696
9	5.569	5.5712
10	0.1	0.1
Optimal volume (in^3)	15931.8	15935.75

Table 3. Loading conditions for 25-bar truss problem

Loading conditions	Node	Direction x (in kips)	Direction y (in kips)	Direction z (in kips)
1	1	0.5	0.0	0.0
	2	0.5	0.0	0.0
	3	1.0	10.0	-5.0
	4	0.0	10.0	-5.0
2	2	0.0	-100.0	-5.0
	4	0.0	-100.0	-5.0

Table 4. Results of the 25-bar transmission tower example

Fixed move-limits	10%	25%	35%
Number of iterations	35*	45	No convergence
Variable move-limits	25% - 10%	35% - 10%	40% - 10%
Number of iterations	24	20	No convergence

Table 5. Comparison of results for 25-bar truss problem

Member No.	Reference (in^2) area	Results (in^2) area
1	0.01	0.01
2,3,4,5	2.0476	1.9581
6,7,8,9	2.9965	2.8273
10,11,12,13	0.01	0.01
14,15,16,17	0.6853	0.7069
18,19,20,21	1.6217	1.7919
22,23,24,25	2.6712	2.6426
Optimal volume (in^3)	5450.4	5455.1

The variable move-limits procedure expanded this range to 35% maximum move. Even if we had selected the 25% maximum move that was used in the fixed move-limit case, the new method required less iterations to reach the optimum.

The method was tested on two additional truss problems with stress and displacement and stress constaints and performed according to expectations. It is presently being tested on other optimization problems, specifically plate problems with aerodynamic and vibrational constraints.

6 Conclusion

This paper presents a method to evaluate move-limits automatically during the optimization process. It depends on the exponents that are computed for the two-point exponential approximation. These exponents are a measure of non-linearity and can be correlated to the individual move-limits of each design variable. The method is implemented and tested on several standard structural optimization cases with stress and displacement constraints. The results obtained show that the method often reduces the number of iterations required to reach the optimum. The method still relies on the engineer to set the maximum and minimum allowable move-limits which are usually derived from experience and problem type. It then tightens the move-limits or relaxes them based on the non-linearity of the functions. The method is also shown to occasionally allow the engineer more latitude in selecting the maximum move allowable.

References

Bathe, K.J.; Wilson, E.L. 1976: *Numerical methods in finite elements.* Englewood Cliffs: Prentice Hall

Bloebaum, C. 1991: Formal and heuristic system decomposition methods in multidisciplinary synthesis. *NASA CR-4413*

Fadel, G.M.; Riley, M.F.; Barthelemy, J.F.M. 1990: Two point exponential approximation method for structural optimization. *Struct. Optim.* **2**, 117-124

Haug, E.J.; Arora, J.S. 1979: *Applied optimal design - mechanical and structural systems.* New York: Wiley InterScience

Pritchard, J.I.; Adelman, H.M. 1990: Differential equation based method for accurate approximations in optimization. *NASA TM-102639, AVSCOM Technical Memorandum 90-B-006*

Sareen, A.K.; Fadel, G.M.; Shrage, D.P. 1991: Application of a two point exponential approximation method in optimizing rotorcraft airframe structures. *Proc. American Helicopter Society, Specialists meeting on Rotorcraft structures* (held at NASA, Langley)

Swanson Analysis Systems Inc. 1990: *ANSYS, engineering analysis system. User's Manual. Vols. I and II*

Vanderplaats, G.N. 1973: CONMIN, a Fortran program for constrained function minimization. User's Manual. *NASA TM-X-62, 282*

Received Nov. 10, 1992
Revised manuscript received June 23, 1993
Communicated by Dr. J. Sobieski