

## Design Sensitivity Analysis

---

Design sensitivity analysis is used to obtain the sensitivity of various aspects of a design model with respect to changes in design parameters in order to facilitate structural modifications. The design parameters that are amenable to change are called “design variables”. The two major aspects of the design model for which design sensitivity is considered herein are:

- a. Design objective
- b. Design model response

As a result, the design sensitivity analysis capability in MSC.Marc is currently limited to finite element models of structures with linear response in the computation of

1. Gradients for
  - a. An objective function (or the design objective), if one is defined by you (for example, minimizing the material volume in the model).
  - b. Various types of design model responses under multiple cases of static mechanical loading, or free vibration, in linear behavior.
2. Element contributions to the responses of the model.

The gradient of the objective function or of a response function is simply the set of derivatives of such a function with respect to each of the design variables, at a given point in the design space (that is, for a given design).

For sensitivity analysis to proceed, the design model, the analysis requirements, the design variables, and the functions for which the gradients are to be found have to be specified by you.

The existing design sensitivity analysis capability in MSC.Marc can be applied in one of two ways:

1. As a stand-alone feature, where you are concerned only with obtaining sensitivity analysis results. Such an application is completed with the output of the sensitivity analysis results.
2. Within a design optimization process, where you are concerned mainly with the optimization of a design objective related to a finite element model. This type of an application of sensitivity analysis is transparent to you. The design optimization process is completed with the output of

design optimization related data, such as the optimized objective function, related values of the design variables, and the analysis results for the optimized design.

These two procedures are described below.

1. Sensitivity analysis as a stand-alone feature

This type of solution usually aims at obtaining the derivatives of user prescribed response quantities at a given design, with respect to each of multiple design variables specified by the user. This set of derivatives is therefore the gradient of the response function at the given design in the design variable space (or, in short, in the design space). For example, for a prescribed response function  $r$ , given the design variables  $x_1$ ,  $x_2$ , and  $x_3$ , the gradient is defined as

$$\nabla_{\tilde{r}} = \left[ \frac{dr}{dx_1} \quad \frac{dr}{dx_2} \quad \frac{dr}{dx_3} \right]^T \tag{5-214}$$

The number of response quantities for which gradients are computed is limited either by the program default or by a user-specified number. If you are interested in obtaining the sensitivity analysis results in order of criticality, the option to sort them in this order is also available. The responses are currently prescribed as constraints with user-defined bounds. If sorting is not required, the bounds can be mostly arbitrary, although they still have to conform to the type of constraint prescribed. However, if sorting is required and is to be meaningful, it is important for you to give realistic bounds on the response.

Element contributions to each response quantity are obtained as a by-product of the type of response sensitivity analysis capability in MSC.Marc. Thus, the response  $r$  can be represented as a sum of these element contributions:

$$r = \sum_{e=1}^N r_e + C \tag{5-215}$$

where the second term,  $C$ , involves work done elsewhere, such as in support settlement, if any.

This is helpful for a visual understanding of which regions of the model contribute in what manner to each of the response quantities at the given design, since it can be plotted in a manner similar to, say stress contours.

Finally, as an option, if you also prescribe an objective function, the gradient of the objective function with respect to the design variables is also computed at the given design. Thus, for the objective function  $W$ , MSC.Marc obtains

$$\nabla_{\tilde{W}} = \left[ \frac{dW}{dx_1} \quad \frac{dW}{dx_2} \quad \dots \quad \frac{dW}{dx_n} \right]^T \quad (5-216)$$

## 2. Sensitivity analysis within a design optimization process:

The design optimization algorithm in MSC.Marc requires the utilization of gradients of the objective function and of the constraint functions, which are very closely related to the response functions. The current algorithm, described under “design optimization”, ignores your initial prescribed design, but instead begins by generating other designs within the prescribed bounds for the design variables. Once the optimization algorithm is completed and the optimized design is available, if a sensitivity analysis is required at the optimized state, it will be necessary for you to modify the model accordingly and to use sensitivity analysis as a stand alone feature.

During design optimization, sensitivity analysis is performed for a maximum number of constraints either indicated by the program default or prescribed by you.

## Theoretical Considerations

The method currently employed in MSC.Marc for response sensitivity analysis is the “virtual load” method. For sensitivity analysis of the objective function, finite differencing on the design variables is performed directly.

In the virtual load method, first a design is analyzed for the user-prescribed load cases, and, if also prescribed, for eigenfrequency response. The response of the structure having been evaluated for each of these analyses, the response quantities for which sensitivity analysis is to be performed are then decided upon and collected in a database. In sequence, a virtual load case is generated for each such response quantity. Reanalysis for a virtual load case leads to virtual displacements.

The principle of virtual work is then invoked. This defines the element contributed part of the response quantity, for which the virtual load was applied, as a dot product of the structural displacement vector, for the actual load case with which the response is associated, and the virtual load vector itself.

The  $j^{\text{th}}$  response  $r_j$  can be expressed as the dot product of the actual load vector with the virtual displacement vector as

$$r_j = \mathbf{p}^T \mathbf{u}_v \quad (5-217)$$

By differentiation of this expression, you can show that the derivative of the response  $r_j$  with respect to a given design variable  $x_i$  is given by

$$\frac{dr_j}{dx_i} = \frac{dp_v^T}{dx_i} \mathbf{u} + \mathbf{u}_v^T \left( \frac{d\mathbf{p}}{dx_i} - \frac{d\mathbf{K}}{dx_i} \mathbf{u} \right)$$

where  $\mathbf{K}$  is the stiffness matrix of the structure. The response derivative above is evaluated on the element basis as:

$$\frac{dr_j}{dx_i} = \sum_e \left[ \frac{dp_v^T}{dx_i} \mathbf{u} + \mathbf{u}_v^T \left( \frac{d\mathbf{p}}{dx_i} - \frac{d\mathbf{K}}{dx_i} \mathbf{u} \right) \right]_e \quad (5-218)$$

where the vectors  $\mathbf{u}$  and  $\mathbf{p}_v$  are the vector of element nodal displacements due to the actual load case and the vector of element nodal forces due to the virtual load, respectively. The case of eigenfrequencies follows the same logic except that an explicit solution for the virtual load case is not necessary. The derivatives are now evaluated at the element level via finite differencing. This is known as the semi-analytical approach.

Note that the derivative expression for the virtual load method is quite similar to that for the Adjoint Variable method. In fact, while they are conceptually different approaches, for certain cases they reduce to exactly the same expressions. However, for certain other cases, the terms take on different meanings although the end result is the same.

## Design Optimization

---

Design optimization refers to the process of attempting to arrive at certain ideal design parameters, which, when used within the model, satisfy prescribed conditions regarding the performance of the design and at the same time minimize (or maximize) a measurable aspect of the design. In MSC.Marc, you can ask to minimize

1. total material volume
2. total material mass
3. total material cost.

When there is more than one material in the finite element model, the specification of different mass densities and unit costs are taken into account in the computation of the objective function.

The performance requirements might not necessarily have to be related to response, but also to different concepts such as packaging, design envelope, even maintenance. The current capability is based on optimization with constraints on response. Also, the lower and upper bounds on the design variables themselves define the limits of design modifications. Hence, the design optimization problem can be posed mathematically in the following format:

$$\begin{array}{ll} \text{Minimize} & W \\ \text{Subject to} & c_j \geq 0.0 \quad j = 1, \dots, m \end{array}$$

where  $W$  is the objective function, and  $c_j$  is the  $j$ th constraint function, either specified as an inequality related to a response quantity or as a limit on a design variable. For example, to limit the  $x$ -direction translation component at a node  $k$ , the constraint can first be written as

$$(u_x)_k \leq u_{xk}^*$$

Assuming that the displacement is constrained for positive values, the normalized constraint expression  $c$  (dropping the subscript  $j$ ) becomes:

$$(u^* - u) / u^* \geq 0.0 \tag{5-219}$$

with its derivatives as:

$$\frac{dc}{dx} = \frac{-1}{u^*} \frac{du}{dx} \tag{5-220}$$

Within MSC.Marc, the constraints can be imposed on strain, stress, displacement, and eigenfrequency response quantities. For stresses and strains, the constraints are defined as being on the elements, and for displacements, the constraints are defined as being at nodes. Stress and strain components, as well as various functions of these components (the von Mises equivalent stress and principal stresses, stresses on prescribed planes) and generalized stress quantities can be constrained. Similarly, translation and rotation components of displacement, resultant and directed displacements as well as relative displacements between nodes can be constrained. For free vibration response, constraints can be placed on the modal frequencies as well as on differences between modal frequencies. A full listing of such constraints are given in *MSC.Marc Volume C: Program Input*.

The upper and lower bounds on the design variables are posed as

$$x_i \leq x_u \quad \text{and} \quad x_i \geq x_l \quad (5-221)$$

after which they can be transformed into expressions similar to [Equation 5-219](#).

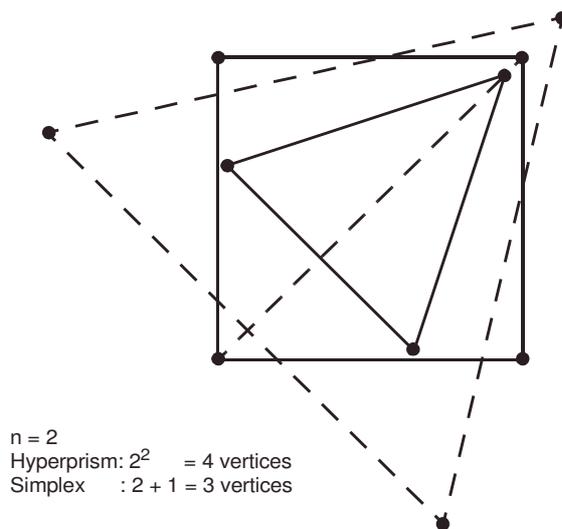
The response quantities associated with the model are implicit functions of the design variables. Analyses at sample design points are used to build explicit approximations to the actual functions over the design space. This approach minimizes the number of full scale analyses for problems which require long analysis times such as for nonlinear behavior. This method is summarized next.

### Approximation of Response Functions Over the Design Space

The design space for the optimization problem is bounded by limits on the design variables of a model to be optimized. The simplest case is that of a single design variable, where the design space is a straight line, bounded at the two ends. For higher number of design variables, say  $n$ , the design space can be visualized as a bounded hyperprism with  $n^2$  vertices. For such a construct, you can build approximations to the constraint functions by way of analyses conducted at the vertices. However, this requires  $n^2$  analyses.

We now note that the minimum geometrical construct spanning  $n$ -dimensional space is a simplex with  $n + 1$  vertices. Thus, an approximation based on analyses at vertices requires only  $(n + 1)$  analyses. The simplex has already been used for first order response surface fitting based on only function values [Reference 3](#). However, the use here involves higher order response functions.

Like the hyperprism, in one-dimension, the simplex degenerates into a straight line. However, in two dimensions it is a triangle, and in general it is a hyper-tetrahedron. [Figure 5-21](#) compares the simplex to the hyperprism in normalized two-dimensional design space.

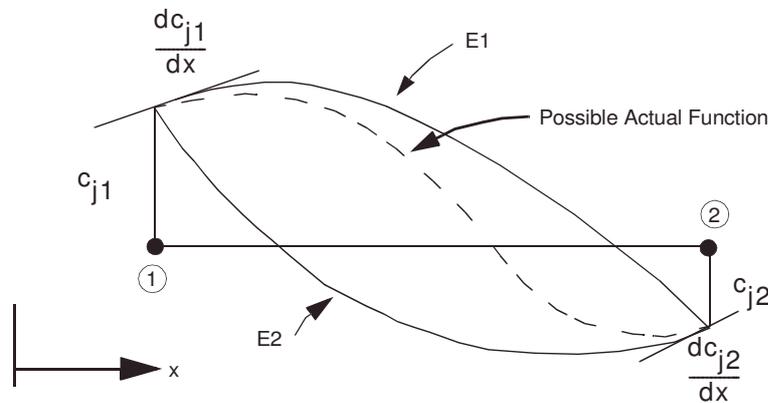


**Figure 5-21** Comparison of the Simplex to the Hyperprism in Two-dimensions

While the orientation of the simplex in the design space appears to be a relatively arbitrary matter, once an origin and the size of the simplex are prescribed, a simple formula will locate all vertices of a simplex in n-dimensions [Reference 4](#). The response gradient information at the simplex vertices is combined with the function values to achieve enhanced accuracy. Thus, an analysis at a vertex can be utilized to yield both response function values and, by way of sensitivity analysis, the response gradients at that vertex. The virtual load method employed in MSC.Marc for obtaining the response gradients is discussed under “[Design Sensitivity Analysis](#)”. The response gradients can then easily be converted to constraint gradients for use in an optimization algorithm.

As a result, the results of an analysis at a vertex of the simplex can be summarized as the vector of constraint function values  $c_j$ , and the gradient,  $\nabla c_j$ , of each constraint function ( $j$ ) at that vertex.

For the case of a one variable problem, the results of analyses at the two vertices are symbolized in [Figure 5-22](#), for a hypothetical constraint  $c_j$ .



**Figure 5-22** Vertex Results for One-dimensional Design Space

From Figure 5-22, it appears almost natural to fit a cubic function to the four end conditions (two function values and two slopes) depicted in the figure. However, this approach is too rigid, and is not easily generalizable to higher dimensions. On the other hand, the use of two quadratics, which are then merged in a weighted manner gives higher flexibility and potential for increased accuracy.

It can be seen that the two equations,  $E1$  and  $E2$ , are designed such that they both satisfy the function values at the two vertices, but  $E1$  satisfies the slope at vertex 1 only, and  $E2$  satisfies the slope at vertex 2 only. Finally, at any design point  $x$ , the response function  $c_j$  can be approximated as

$$c_j = (W1 E1 + W2 E2) / (W1 + W2) \quad (5-222)$$

where the weight functions  $W1$  and  $W2$  are normally functions of  $x$  and possibly some other parameters.

This type of approach has the further advantage that it is consistent with the use of the simplex for determining analysis points and approximating constraint functions in the higher dimensional cases. Therefore, for an  $n$ -dimensional problem, the simplex having  $n + 1$  vertices, each equation needs to have  $(n + 1) + n = 2n + 1$  unknown parameters. The general quadratic polynomial without the cross-coupling terms satisfies this condition for all  $n$ .

## Improvement of the Approximation

When approximations are used, the results of an optimization process need to be checked by means of a detailed analysis. As a result, the approximations can be adjusted and the optimization algorithm can be reapplied. Depending on how accurate the approximations prove to be and how many more detailed analyses are acceptable to you, this process can be applied for a number of cycles in order to improve upon the results.

## The Optimization Algorithm

The optimization algorithm implemented in MSC.Marc is the Sequential Quadratic Programming method [Reference 5](#). This method is employed using the approximation described above. By obtaining response function and gradient values from the approximate equations, the need for full scale analyses is reduced. The method is summarized below.

The quadratic programming technique is based on the approximation of the objective function by a quadratic function. When nonlinear constraints exist, as is the case in most practical design optimization problems, the second order approximation concept is extended to the Lagrangian which is a linear combination of the objective function and the constraint functions. The solution method for a quadratic programming problem with nonlinear constraints can be summarized as the following:

At each step, modify the design variables vector  $x$  by adding a scaled vector

$$x \leftarrow x + qs \quad (5-223)$$

where  $s$  is a search direction and  $q$  is the scale factor along the search direction. If the search direction has been determined, the scale factor can be found by some type of line search along the search direction. The determination of the search direction constitutes the major undertaking in the quadratic programming method. If  $H$  is the Hessian of the Lagrangian and  $g$  is the gradient of the objective function, then the search direction  $s$  is that which minimizes the function

$$g^T s + (s^T H s) / 2 \quad (5-224)$$

subject to the linearized constraints

$$J s \geq -c \quad (5-225)$$

where  $J$  is the Jacobian matrix of the constraints and  $c$  is the vector of constraint functions at the current iteration. Due to lack of knowledge about which constraints will be active at the optimum, the Hessian of the Lagrangian is not always readily available. Thus, some iterations take the form of a gradient projection step.

The coefficients of the constraint functions in the Lagrangian are the Lagrange multipliers which are unknown before solution has started. At the optimum these multipliers are zero for inactive constraints. Normally, the above problem is solved using only those constraints which appear to be active at or close to the current design point, with the assumption that these constraints will be active at the optimum. The selection of these active constraints is done within the framework of an active set strategy, the set being modified appropriately with the progression of the iterations. Similarly, the arrays  $g$ ,  $H$ , and  $J$  are also modified as the iterations proceed.

## MSC.Marc User Interface for Sensitivity Analysis and Optimization

This is discussed below under input, output, and postprocessing.

### Input

Input features related to design sensitivity and design optimization are similar. However, they differ in the parameter data blocks and in the optional specification of an objective function for the case of sensitivity analysis. Therefore, other than these two items, all discussion of input should be construed to refer to both procedures equally. All design sensitivity and design optimization related parameters and options in a MSC.Marc input file start with the word `DESIGN`.

All load cases and any eigenfrequency analysis request associated with sensitivity analysis or optimization should be input as load increments in the history definition section after the `END OPTION` line. The maximum number of nodes with point loads or the maximum number of distributed load cases should be defined in the `DIST LOADS` parameter.

The type of solution requested from MSC.Marc can simply be indicated by the parameter input `DESIGN SENSITIVITY` or `DESIGN OPTIMIZATION`. Only one of these lines can appear in the input. These parameters also let you change the sorting of constraints, the maximum number of constraints in the active set, and the maximum number of cycles of design optimization. The first two items are discussed in previous sections. A cycle of design optimization refers to a complete solution employing the sequential quadratic programming technique followed by a detailed

finite element analysis to accurately evaluate the new design point reached by way of approximate solution. If required, the approximation is improved using the results of the last finite element analysis, and a new cycle is started.

The specification of an objective function, being optional for design sensitivity, is made by use of the model definition option **DESIGN OBJECTIVE**. This allows you to choose from one of the available design objectives.

The **DESIGN VARIABLES** option allows you to specify the design variables associated with the finite element model. You have a choice of three major types of design variables:

1. Element cross-sectional properties as given by way of the **GEOMETRY** option.
2. Material properties as given by **ISOTROPIC** or **ORTHOTROPIC** options.
3. Composite element properties of layer thickness and ply angle, as given by the **COMPOSITE** option.

The properties which are supported are listed under the **DESIGN VARIABLES** option. The relevant elements, for which cross-sectional properties can be specified as design variables, each has a section in *Volume B: Element Library*, describing which properties, if any, can be posed as design variables for that element.

Similarly, the list of material properties (currently all related to linear behavior) that can be design variables is given under the **DESIGN VARIABLES** option. For composite groups, the layer thicknesses and ply angles can be given as design variables for composite groups.

Design variables can be linked across finite element entities such that a given design variable controls several entities. An example is the linking of the thicknesses of several plane stress elements by means of a single design variable. Thus, when this variable changes, the thicknesses of all linked elements reflect this change. On the other hand, for the unlinked case, all thicknesses are associated with separate design variables. This feature is controlled by the **LINKED** and **UNLINKED** commands.

Response quantities, or constraints on response quantities, are specified by means of the **DESIGN DISPLACEMENT CONSTRAINTS**, **DESIGN STRAIN CONSTRAINTS**, **DESIGN STRESS CONSTRAINTS**, and **DESIGN FREQUENCY CONSTRAINTS** options. There is no limit to the number of constraints. Displacement constraints are posed at nodes or groups of nodes, while the strain and stress constraints are posed over

elements or groups of elements, and frequency constraints are posed for free vibration modes. A complete list can be found under the above mentioned options in *MSC.Marc Volume C: Program Input*.

When strain or stress constraints are prescribed, it is useful to know that the program evaluates such constraints at all integration points of all layers of an element and proceeds to consider the most critical integration point at the related layer for the element. Thus, a strain or stress constraint on an element normally refers to the most critical value the constraint can attain within the element. During optimization, the most critical location within the element may change and any necessary adjustment takes place internally.

For certain responses, the limiting values can be the same in absolute value for both the positive and negative values of the response. For constraints on such response functions, you have the choice of prescribing either separate constraints on the positive and negative values, or a combined constraint on the absolute value. The first approach is more accurate albeit at a higher computational cost.

### Output

For sensitivity analysis, the output file contains the following information:

- Echo of input, any warnings or error messages.
- A of numbers and definitions for your prescribed design variables. It is important to note that the variable values are always output in the internal numbering sequence which is defined in this. Search for the words 'Design variable definitions' to reach this in the output file.
- Analysis results for your prescribed design.
- The value of the objective function and its gradient with respect to the design variables prescribed by you.
- Sensitivity analysis results for the response functions in the default or user-defined set, sorted in order of criticalness (if specified). It should be noted that although the responses are sorted across multiple load cases, the sensitivity results are output for each load case. The related output file consists of a check on the actual response value (obtained by sensitivity analysis) and the gradient of the response with respect to the design variables. For eigenfrequency results, the check values on the response can be somewhat more accurate than the results from eigenfrequency analysis since the latter is iterative, but the check uses the Rayleigh quotient on top of

the iteration results. A constraint reference number allows you to track the sensitivity analysis results plots when postprocessing. Search for the word 'Sensitivity' to reach this output.

For design optimization, the following is written into the output file:

- Echo of input, any warnings or error messages.
- Certain indications that some analyses are being done, but no analysis results except for the 'best' design reached during optimization.
- The objective function values and the vector of design variable values:
  - At each of the simplex vertices.
  - For the starting simplex vertex (from this point on, the information also includes whether the design is feasible or not).
  - At the end of each cycle of optimization.
  - For the 'best' design found.
- Analysis results for the 'best' design.

### **Postprocessing**

This requires that you ask the program to create a post file. The following plots can be obtained by way of postprocessing.

For sensitivity analysis:

- Bar charts for gradients of response quantities with respect to the design variables.
- Contour plots of element contributions to response quantities. Thus, a finite element model contour plot gives the element contributions to a specific response quantity which was posed in the form of a constraint in the input file.

The increment number of the sensitivity analysis results is the highest increment number available in the post file. The information for each response quantity is written out as for a subincrement. The zeroth subincrement corresponds to the objective function information. Numbers for the other subincrements correspond to the constraint reference number(s) in the output file.

For design optimization:

- Path plots showing the variation of the objective function and of the design variables over the history of the optimization cycles. The best design (feasible or not) is not necessarily the last design point in the plot. The values at the starting vertex are considered as belonging to the zeroth subincrement. Each

optimization cycle is then another subincrement. The increment number corresponding to these subincrements is taken as zero. The analysis results for the 'best' design start from increment one.

- Bar charts where each chart gives the values of the design variables at the optimization cycle corresponding to that bar chart.