

What-If Post-Processing of General-Purpose Optimization Results Using VisualDOC

Gerhard Venter (gventer@vrand.com) *

Gary C. Quinn (quinng@vrand.com) †

Vladimir Balabanov (vladimir@vrand.com) ‡

Garret N. Vanderplaats (vanderplaats@vrand.com) §

Vanderplaats Research and Development, Inc.

1767 S 8th Street, Colorado Springs, CO 80906

This paper introduces and demonstrates a new post-processing capability that is available in Version 4.0 of VisualDOC. VisualDOC is a general-purpose optimization tool, available from Vanderplaats Research and Development, Inc. This new post-processing tool allows the user to perform real-time “What-If?” studies in the vicinity of the optimum point found by the optimizer. The tool is demonstrated using a stiffened panel that requires a non-linear analysis.

I. Introduction

DESPITE many years of research, resulting in the availability of several general-purpose optimization programs, optimization has only realized limited success in the industrial environment. There are many reasons for this lack of acceptance, including (a) a lack of user familiarity with optimization concepts, and (b) the high computational resource requirements for general-purpose optimization. The first issue is primarily due to the fact that optimization is rarely taught at the undergraduate level, creating the need for user training that companies are often unwilling to invest in. The high computational resource requirements stems from the general-purpose requirement placed on these optimization algorithms. In most cases general-purpose optimization makes use of gradient based optimization algorithms, primarily due to their high efficiency. Unfortunately, gradient information is not readily available for most engineering analyses, requiring general-purpose optimizers to employ finite difference gradient calculations. These finite difference gradient calculations are the main cost component for most general-purpose optimizers.

To address these issues, vendors have created powerful, general-purpose optimization software that are easy to use. The VisualDOC program,¹ from VR&D is one example of such a commercial quality optimization code. VisualDOC addresses the lack of familiarity with optimization concepts issue by providing the user with a powerful and robust optimization engine, an easy to use GUI and a design integration tool for quick and easy coupling to existing analysis programs. To reduce computation cost, VisualDOC provides the user with efficient optimization algorithms that allow many calculations, including finite difference calculations, to be executed concurrently. VisualDOC is built on a platform independent database that allows the user to further reduce computational cost by reusing design points that were previously analyzed. The intuitive graphical user interface was specifically designed to guide users through the steps required to setup an optimization problem, allowing engineers to apply optimization to real problems and obtain meaningful results, with minimal training. The optimization engine used in VisualDOC is a robust optimizer that has been used for many years in a wide variety of commercial codes. VR&D has spent significant time to provide the optimization engine with default parameters that will work for most problems, thus further reducing user training requirements.

*VisualDOC Project Manager, AIAA Member

†Senior R&D Engineer

‡Senior R&D Engineer, Senior AIAA Member

§President, AIAA Fellow

The VisualDOC capabilities include gradient based optimization algorithms,² non-gradient based optimization algorithms (genetic algorithm and particle swarm optimization^{3,4}) response surface approximate optimization, design of experiments, integer, discrete and mixed design problems and probabilistic analysis and optimization. VisualDOC is also available as an Application Programming Interface (API) that allows users to embed the technology in their own analysis programs.

An important aspect of general-purpose optimization is the presentation of the results to the user. Traditionally, this includes text reports and design history charts of the design variable, response and objective function values. Although these are powerful tools for investigating the optimization results, they all have one big drawback: the user cannot interact with the results. For example, it may be useful for the user to investigate the influence of small changes to the optimum design point on the objective function value. Being able to quickly investigate the influence of small changes to the optimum design point is an important part of an optimization study since, in practice, a user rarely performs an optimization study only once. Typically, the user performs an optimization study, investigates the optimum results, gains more insight into the problem and makes small modifications to the optimization problem definition to re-run the optimization study. This process of re-running an optimization study many times to perform “What If?” studies about the optimum design point adds to the overall cost of doing optimization.

Version 4.0 of VisualDOC has addressed this need by introducing an unique “What If?” study tool that allows the user to interact with the optimum results. The user can use this tool to investigate the influence of changes to the optimum design variable values and/or constraint values on the optimum objective function. The “What If?” study tool provided with VisualDOC employs a full optimization on an approximate problem to estimate the influence of the changes to the optimum result. This new “What If?” post-processing tool is the focus of this paper.

II. Example Problem

The following example problem will be used throughout this paper to illustrate optimization post-processing capabilities. The example problem was obtained from the ABAQUS Example Problems Manual⁵ and consists of a rectangular plate that is 10.8 *m* long and 6.75 *m* wide with a skin thickness of 5.0 *mm*. The plate is reinforced with several stiffeners in both the longitudinal and transverse directions as shown in Fig. 1. The boundary conditions reflect the fact that the plate represents part of a larger structure. The two longitudinal sides have symmetric boundary conditions, and the two transverse sides have pinned boundary conditions. In addition there are springs at two major reinforcement intersections that represent flexible connections to the rest of the structure. The mesh consists of 1,975 grid points and 2,214 elements. The elements consist of 1,912 S4 shell elements for both the plate and larger reinforcements and 14 S3 shell and 288 B31 beam elements for the remaining reinforcements. The entire structure is made of the same construction steel, with an initial flow stress of 235.0 *MPa* and the finite element model is constructed using meters as the length measurement.

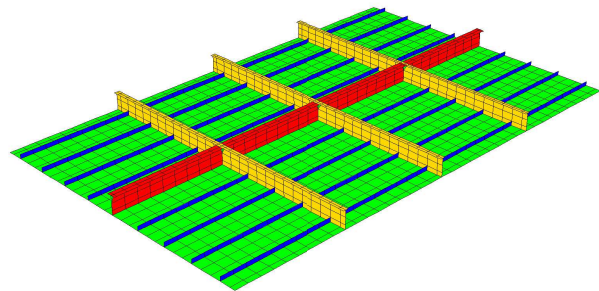


Figure 1. ABAQUS Stiffened plate example problem

The analysis consists of two ABAQUS steps. In the first step a gravity load is applied perpendicular to the plane of the plate. In the second step a longitudinal compressive load of 6.46×10^6 *N* is applied to one of the pinned sides of the plate. All the nodes on that side are forced to move in unison by means of multi-point constraints. The analysis is quasi-static, but buckling occurs. Initially, local out-of-plane buckling develops throughout the plate in an almost checkerboard pattern, inside each one of the sections delimited by the reinforcements. Later, global buckling develops along a front of sections closer to the applied load, as shown in Fig. 2(a). In Fig. 2(a) the contours represent the out of plane displacement. The load-displacement curve obtained by comparing the longitudinal applied load versus the displacement at the side where the load is applied, is shown in Fig. 2(b). Figure 2(b) clearly reflects the global instability that develops. The curve becomes almost flat, indicating a complete loss of load carrying capacity.

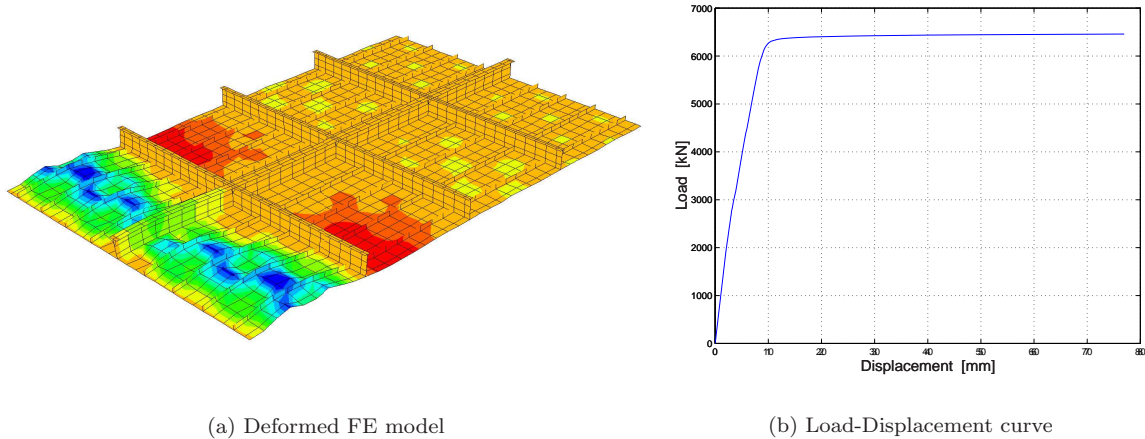


Figure 2. Results for the initial stiffened plate

The optimization problem is defined as the redistribution of the existing mass to avoid the global buckling that is shown in Fig. 2. This is achieved by minimizing the longitudinal displacement where the load is applied, with an upper bound constraint that the mass should not exceed its initial value of 4598.96 kg. Four thickness design variables are considered as shown by the different color regions of Fig. 1. The first (t_1) is the skin thickness (green region of Fig. 1); the second (t_2) is the web thickness of the small longitudinal stiffeners (blue region of Fig. 1); the third (t_3) is the web and cap thickness of the large longitudinal stiffener (red region of Fig. 1); and the fourth (t_4) is the web and cap thickness of the transverse stiffeners (gold region of Fig. 1). The lower and upper bounds for the four design variables are 1.0 mm and 100.0 mm respectively.

The example problem was solved using the response surface approximate optimization module of Visual-DOC. The initial point was chosen close to that considered in the original ABAQUS example problem, with $t_1 = 5.0$ mm, $t_2 = 6.0$ mm and $t_3 = t_4 = 10.0$ mm. The optimizer was able to solve this problem using a total of 34 analyses. The optimum solution has a weight of 4592.86 kg (the initial weight was 4598.96 kg) and a displacement of 7.96 mm (the initial displacement of 76.97 mm). The longitudinal displacement was thus reduced by a factor of 10 and the global buckling is clearly avoided as shown in the deformation and load-displacement plots of Fig. 3. The optimum design point is located at $t_1 = 5.3$ mm, $t_2 = 18.2$ mm and $t_3 = t_4 = 1.0$ mm.

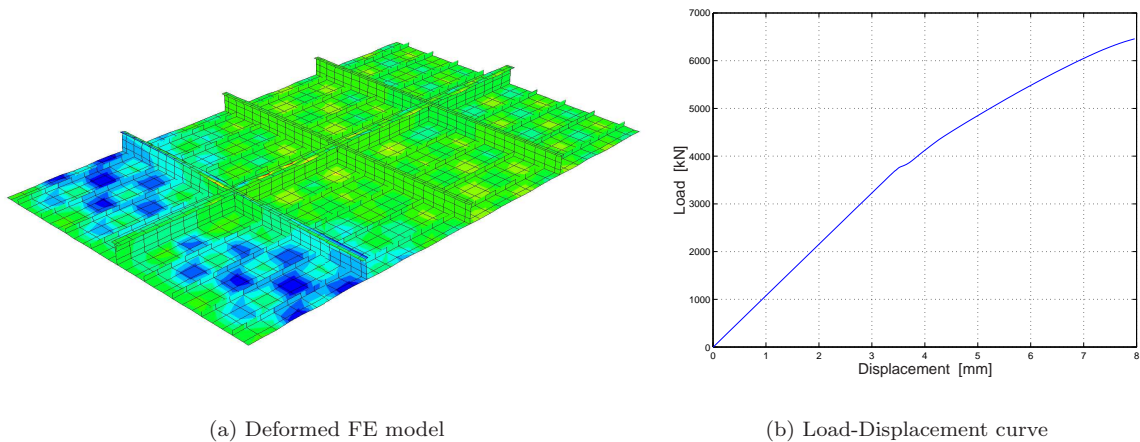


Figure 3. Results for the optimized stiffened plate

III. Traditional Post-Processing

Traditionally, general-purpose optimization results are presented to the user as either a text report and/or a set of design iteration history charts. These traditional post-processing options are explained in more detail for our example problem next, with special emphasis on the post-processing options that are provided in the VisualDOC program.

A. Text Reports

A text report typically includes a summary of the design points visited during the optimization run, the objective function and constraint values evaluated at each design point and an explanation for the convergence of the algorithm. For gradient based optimization, more detailed reports may include gradient information, information regarding the one-dimensional search and Lagrange multiplier data.

The text reports provided in VisualDOC are created on the fly from information stored in the VisualDOC database. This allows the user to customize the information shown in the report, depending on the user's current need. For example, the user can choose to show only a summary of each design iteration or show the design iteration with the one-dimensional search information and to show all constraint values or only constraints that are active or violated.

Typically a VisualDOC report consists of a summary of the optimization algorithm, the algorithm parameters and the optimization problem definition. The summary information is followed by a detailed report for each design iteration that consists of the best point found so far and the design variable, objective and constraint values evaluated during the iteration. Finally, the report also contains information about the number of analysis calls and the overall time it took to complete the optimization. Additionally, the user can obtain more information by turning on output from the optimizer itself. This will include detailed information regarding the one-dimensional search including step sizes, the gradient calculations and the Lagrange multipliers.

VisualDOC allows the user to search through the text report, to cut, copy and paste, to print or to export the report to a text file.

B. Graphical Reports

The traditional graphical reports typically consist of design history charts that show the design iteration history of the objective function, design variables and constraints. In VisualDOC all of these charts are available in real time while the optimization is running, or from the VisualDOC database any time after the optimization is completed. VisualDOC creates these charts in terms of the best design point found to date versus the design iteration. Depending on the algorithm used, a design iteration in VisualDOC may represent anywhere from one (for response surface approximate optimization) to hundreds or even thousands of points (for non-gradient based optimization).

The objective function design iteration history chart for our example problem is shown in Fig. 4.

Note that the chart shows the best objective function versus the design iteration number. The chart also includes information about the feasibility of the design. If one or more constraints are violated, we have an infeasible design, which is indicated with a red star (see the first point in Fig. 4). As the design becomes feasible the red star is replaced with a green dot. The objective function design history shows that the approximate optimization algorithm quickly reached a feasible solution. After increasing the accuracy of the approximation by adding more data (design iterations 3 through 11), VisualDOC was not able to improve the optimum found in integration 2.

The corresponding design variable history chart is shown in Fig. 5. VisualDOC allows the user to also show the design variable side constraints. VisualDOC will never evaluate any design variable outside the

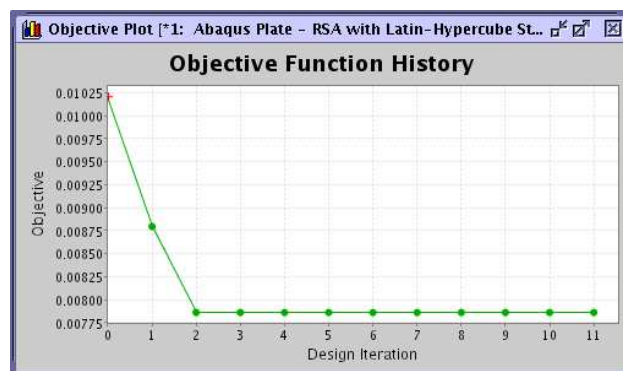


Figure 4. VisualDOC objective function history plot

side constraints, but showing the bounds is useful to indicate any design variables that may be restricted from moving. The corresponding response history chart is shown in Fig. 6. Again the user can choose to show the constraint bounds. In this case the **Mass** response initially exceeded the upper bound. However, the optimizer quickly overcomes this initial violation and at the optimum, the constraint bound is active, meaning that the **Mass** response is within a small tolerance of the allowable upper bound value.

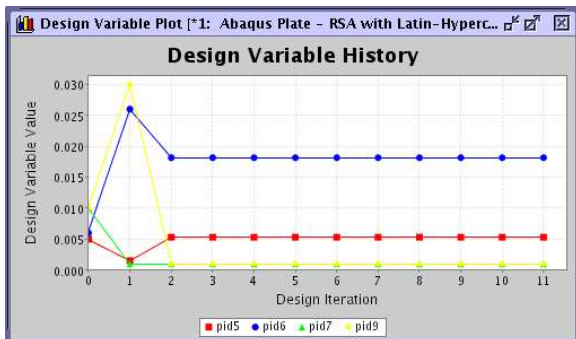


Figure 5. VisualDOC design variable history plot

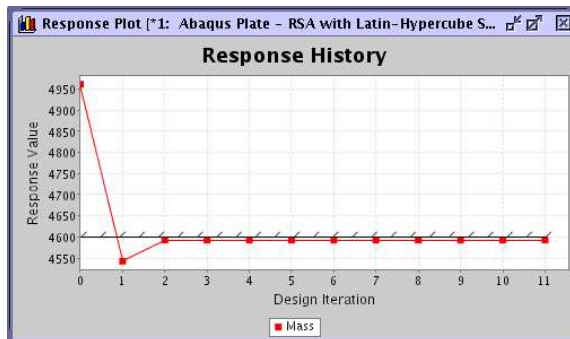


Figure 6. VisualDOC response history plot

VisualDOC allows the user to print, save or modify the charts. For example, the user can change colors, font sizes, the legend, etc. The user can also place the mouse pointer on any point in any of the charts to see the exact value for that point. Additionally, the user can save the data from these charts into a text file. This allows users to create their own charts using any available plotting program.

IV. “What If?” Post-Processing

As shown in the previous section, the traditional text and graphical reporting of optimization results provide a powerful tool for analyzing the optimum results while the optimization is running or after it is completed. However, these tools lack an important aspect; they can only report on what happened without interaction from the user that would influence the reported results.

VisualDOC Version 4.0 provides a unique feature that allows the user to perform what-if studies. This tool allows the user to interact with the optimum design variable values and constraint bounds and perform an approximate optimization based on these changes. This approximate optimization is a full optimization performed on an approximate problem and as a result automatically takes account of interactions between the design variables. This is different from performing a sensitivity analysis that does not account for the interactions between design variables. The tool provided in VisualDOC allows the user to plot a selected sub-set of design variables and responses as shown in Fig. 7. In Fig. 7 we choose to plot all design variables and responses for our example problem. The tool also includes a plot for the objective function and the worst constraint values. In these plots, **pid5** denotes t_1 , **pid6** denotes t_2 , **pid7** denotes t_3 and **pid9** denotes t_4 while **N329(U1)** denotes the longitudinal displacement at the side where the load is applied and **Mass** denotes the overall mass of the structure.

As shown in Fig. 7, the **What If Post-Processing** dialog consists of three sub-plots. These are (1) the design variables sub-plot; (2) the responses sub-plot and (3) the objective function sub-plot. Each of these plots will be discussed in more detail next. In addition, the tool presents the user with four buttons. The **Close** button is used to close the dialog. The **Optimize (Approx.)** button is used to perform an approximate optimization and update the design variables, responses and objective function sub-plots accordingly. This button is only enabled after some changes to either the design variable values and/or the constraint bounds have been made. The **Validate** button allows the user to perform a real analysis for the current design variable and constraint values and update the responses and objective function sub-plots accordingly. This button is also only enabled after some changes have been made to the problem definition. Finally, the **Create VisualDOC Task** button can be used to create a VisualDOC task that includes the modifications made to the problem definition. The newly created VisualDOC task can be used to create a full, and thus more accurate, optimization.

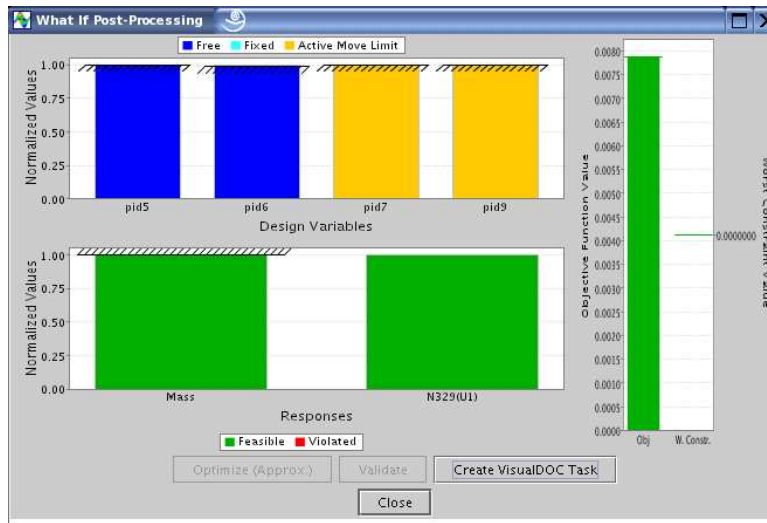


Figure 7. VisualDOC what-if post-processing tool

A. Design variables sub-plot

The design variables sub-plot appears in the top, left hand corner of the **What If Post-Processing** dialog and is reproduced here as Fig. 8. The initial plot shows the design variable values found at the optimum solution, all normalized to 1.00 for bar chart visualization and comparison purposes. The plot also includes upper and lower bound information as shown in Fig. 8.

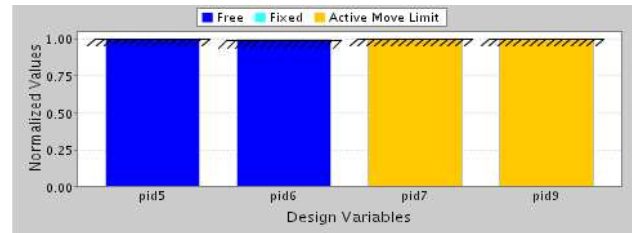


Figure 8. The design variables sub-plot

Initially all bars in the design variables plot are colored a dark blue or orange, which indicate that they are free to be modified by the optimizer. The orange color is used to indicate that a design variable is at a move limit bound and can thus only be changed in one direction. To see the unscaled value of a design variable, simply place the mouse pointer over the bar that represents the appropriate design variable and a popup with the unscaled value will appear. To change a design variable value, simply select the bar to modify and drag the mouse until the desired value is reached. A continuously updated popup will indicate the current value of the design variable as it is modified. Note that a design variable can only be dragged up or down within certain limits. VisualDOC imposes move limits on the design variables to limit the allowed changes that are made. The move limits are required since we are solving an approximate problem that is only accurate within the region of the optimum solution and can be modified by the user. After the mouse is released, the bar representing the design variable will change color from dark blue to light blue. The light blue color indicates that the design variable value was changed and that it will be fixed at this new value during the optimization. Anytime a user changes a design variable value in the what-if post-processing tool, the design variable will be fixed at that new value and cannot be changed by the optimizer when the approximate optimization is performed.

The basic user interaction with the design variables sub-plot include dragging a bar to a new value or placing the mouse pointer over a bar to get the unscaled value. However, there are many more actions that are allowed. To access these additional options, simply select the right mouse button inside the design variables sub-plot. This will provide a popup menu with the following additional options:

- **Set Move Limits:** This option allows the user to change the default move limits imposed by VisualDOC on the design variables. The default move limits are a relative move limit of $\pm 10\%$ and an absolute move limit of ± 0.02 . The relative move limit is always applied, unless it results in a move

limit less than the absolute move limit, in which case the absolute move limit is used.

- **Show All Current Values:** This option allows the user to open a dialog box that displays all the current, unscaled, values for all design variables and responses. The current move limit bounds for the design variables and the constraint bounds are also included.
- **Set Design Variable Value:** This option allows the user to specify a specific value for the currently selected design variable. The new value can be specified as either a scaled or an unscaled value. This action has the same effect as dragging the design variable up or down, except that the move limits are ignored.
- **Free Design Variable:** This option allows the user to free a previously fixed design variable. Design variables are fixed, and cannot be changed during the approximate optimization, whenever the user specifies a new value. When freeing the design, the value of the design variable remains unchanged, but the approximate optimizer will be able to change the value during the optimization.
- **Reset Design Variable:** This option allows the user to reset a design variable back to its initial (optimum) value (normalized to 1.0).
- **Reset All:** This option allows the user to reset all design variable values and constraint bounds back to their initial (optimum) values. Design variables will be normalized to 1.0, constraint lower bounds to -1.0 and constraint upper bounds to 1.0. The objective function and worst constraint plots will also be returned to their initial (optimum) values.
- **Properties:** This option allows the user to control the chart properties.
- **Print:** This option allows the user to print the chart.
- **Save Graph:** This option allows the user to save the chart in one of many different image formats.

B. Response sub-plot

The responses sub-plot appears in the bottom, left hand corner of the **What If Post-Processing** dialog and is reproduced here as Fig. 9. The initial plot shows the response values found at the optimum solution, all normalized so that the upper bound constraint values are at 1.0 and the lower bound constraint values are at -1.0. If no constraints are specified the response itself is normalized to 1.0. The upper and/or lower bound constraints are included in the plot as shown in Fig. 9.

As for the design variables plot, the color scheme in the responses plot is important. A green bar represents a response that satisfies all associated constraint bounds, while a red bar indicates that the response is violating a constraint bound. It is also possible to identify a feasible/infeasible response by comparing the height of the bar to the constraint bounds shown.

To view the unscaled values of the constraint bounds, move the mouse pointer over the desired bound and a popup will appear with the unscaled value for that constraint bound. Note that the unscaled value of the closest bound will be displayed. Thus, if the constraint has both an upper and lower bound set, placing the mouse pointer above the 0 value on the Y axis will result in displaying the unscaled value for the upper bound, while placing the mouse pointer below the 0 value will result in displaying the unscaled value for the lower bound. To modify a bound, simply drag that bound up or down until the desired value is reached. While dragging the bound, a continuously updated popup will display the current unscaled value for that bound. Again, when dragging a bound, the closest bound will be selected for modification. If a response violates a constraint bound due to a modification in the bound value, the response will change from green to red.

The basic user interaction with the responses sub-plot include dragging a bound to a new value or placing the mouse pointer over a bound to get the unscaled value. However, as with the design variables sub-plot

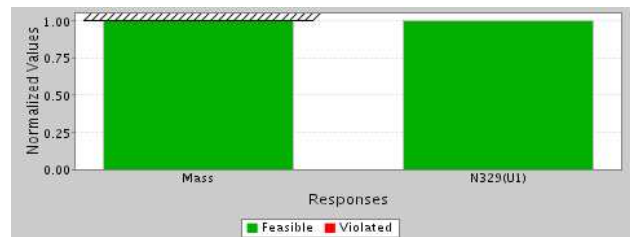


Figure 9. The responses sub-plot

there are many more actions that the user can perform within the responses sub-plot. To access these additional options, simply select the right mouse button inside the responses sub-plot.

C. Objective function sub-plot

The objective function sub-plot is shown on the right hand side of the **What If Post-Processing** dialog and is reproduced here as Fig. 10. This sub-plot consists of two bars, one for the unscaled objective function value and one for the unscaled worst constraint value. These two bars have independent Y-Axis, each with it's own range. The user cannot directly interact with this sub-plot. Instead this plot is automatically updated when the user performs an approximate optimization or a validation of the currently modified optimization problem.

The color scheme in this plot is the same as for the responses plot: green indicates a feasible design (all constraint bounds are satisfied), while red indicates an infeasible design (at least one constraint bound is violated). Also, note that there is a horizontal line drawn at the end of each bar. These lines represent the values of the objective function and worst constraint at the optimum design point and are colored either red or green, indicating a feasible or infeasible optimum design. When performing an approximate optimization or validation, the objective function and worst constraint bounds may change up or down and may change color between red and green. However, the horizontal lines indicating the optimum objective function and worst constraint values will remain, thus providing a reference point to the initial optimum design. The user can use this reference point to investigate the influence of changing the optimization problem definition.

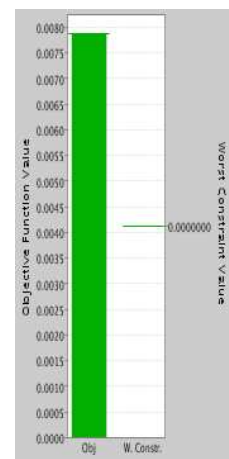


Figure 10. The objective function sub-plot

D. Using the tool

A typical application of the tool may be to quickly investigate changes to the optimum design point, for example to accommodate manufacturing. In our case, let us assume that we can only get the construction steel required to produce the plate in certain thickness. From Fig. 8, the t_3 (indicated by pid7 on the plot) and t_4 (indicated by pid9 on the plot) thickness values are already at their lower bound values of 1.0 mm. We can fix these values at their lower bounds to prevent the optimizer from changing them during the optimization. To do this, simply select the bars with the mouse and drag them

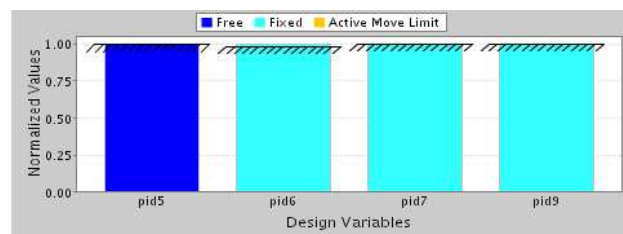


Figure 11. The design variables sub-plot after changing the design variables

all the way down. They will not move (they are already at their lower bounds), but will change color from orange to light blue, indicating that their values are now fixed. Next, we can take a look at the t_2 value (indicated by pid6 on the plot). The optimum value is $t_2 = 18.2$ mm. Let us change this to 15.0 mm by right clicking on the bar and selecting the **Set Design Variable Value** option from the popup menu. The design variables sub-plot should now appear as shown in Fig. 11. Note that due to the small absolute values of the design variables, the changes in the design variable values are not clearly visible in Fig. 11.

With the design variables changed, we can now perform a real-time approximate optimization to find the new approximate optimum value for t_1 . However, because of the small absolute values of the design variables, we first need to adjust our move limits. The optimum t_1 value is 5.3 mm, we want our move limits to be about $\pm 10\%$ so let's change our absolute move limit from 0.02 to 0.0005 (remember that the problem is setup in meters). Next, we can select the **Optimize (Approx.)** button to perform the optimization. Even though a typical analysis takes about half an hour, the approximate optimization result is returned immediately. The what if study tool now appears as shown in Fig. 12. We can clearly see that we have maintained the weight constraint as active, while the displacement has slightly increased. This is to be expected, since we've restricted the optimizer and should thus find a worse design as compared to our original optimization. The new approximate optimum value for t_1 is now 5.04 mm.

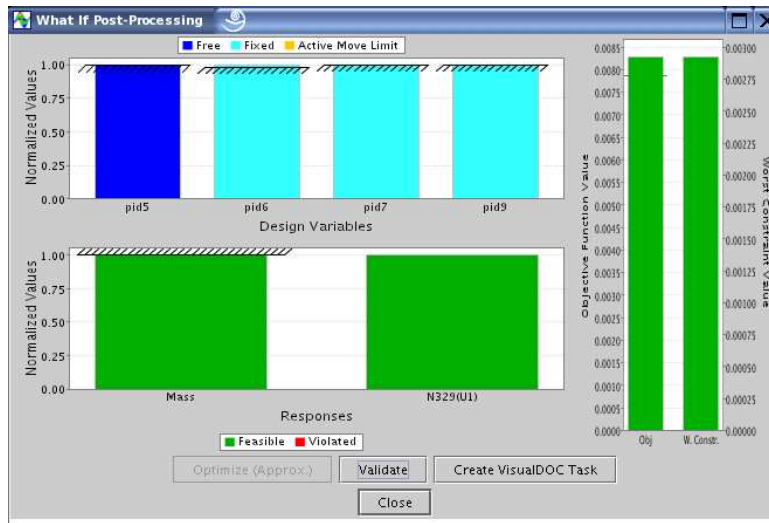
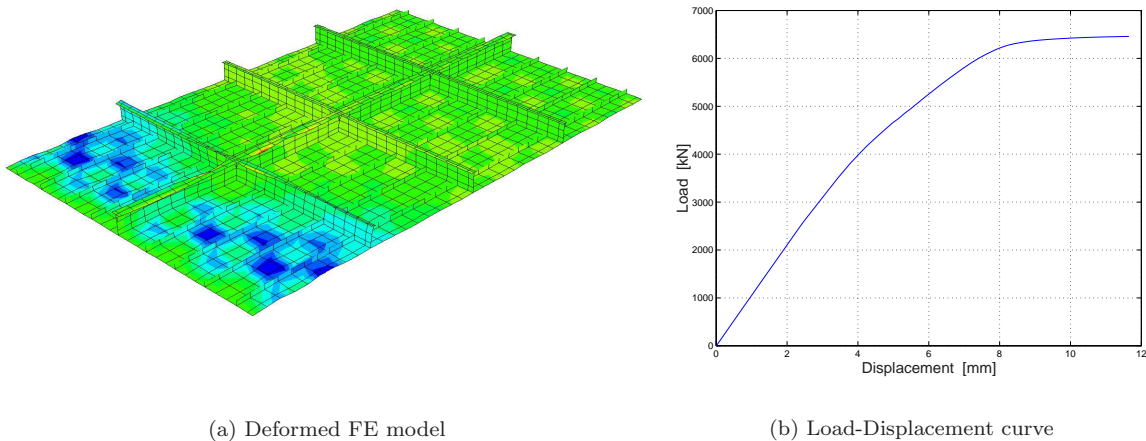


Figure 12. What If Post-Processing dialog after running the approximate optimization

Next, we can fix the value of t_1 at 5.0 mm and rerun the optimization. In this case all the design variables are fixed and we will only obtain an approximate function evaluation at the specified point. For $t_1 = 5.0\text{ mm}$, $t_2 = 20.0\text{ mm}$ and $t_3 = t_4 = 1.0\text{ mm}$ the approximation estimates a displacement of 8.3 mm and a mass of 4558 kg . The final step is to perform a real analysis at this point, by selecting the **Validate** button. The real analysis returned a displacement of 11.6 mm and a mass of 4513 kg . The approximation for the displacement was less accurate than that for the mass.. However, the correct trends were still captured by the what-if study tool, allowing the designer to quickly find a design that is easy to manufacture, based on the original optimum found by the optimizer. The final ABAQUS results for the design selected by the what-if study tool is shown graphically in Fig. 13.



(a) Deformed FE model

(b) Load-Displacement curve

Figure 13. Results for the final stiffened plate

V. Concluding Remarks

A new, interactive post-processing capability is introduced as part of VisualDOC Version 4.0. This capability allows the user to interactively perform real-time what-if studies, based on optimization results. This is especially powerful if the analysis takes a long time. The tool allows the user to estimate the influence of changing the optimization problem in real-time, before committing to a time-consuming optimization. The tool also provides the user with a better feel and more insight into the optimization problem and addresses both the issues of ease of use and high computational cost.

References

¹*VisualDOC Design Optimization Software, Version 4.x, Getting Started Manual*, Vanderplaats Research and Development, Inc., 1767 S. 8th St., Colorado Springs, CO, 2004.

²*DOT. Design Optimization Tools, Version 5.x, Users Manual*, Vanderplaats Research and Development, Inc., 1767 S. 8th St., Colorado Springs, CO, January 2001.

³Venter, G. and Sobieszczanski-Sobieski, J., "Particle Swarm Optimization," *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO*, AIAA-2002-1235, April 22–25 2002.

⁴Venter, G. and Sobieszczanski-Sobieski, J., "Multidisciplinary Optimization of a Transport Aircraft Wing using Particle Swarm Optimization," *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA*, AIAA-2002-5644, September 4–6 2002.

⁵*ABAQUS Example Problems Manual, Version 6.4-3*, ABAQUS, Inc., 1080 Main Street, Pawtucket, RI, 2004.