**11th World Congress on Structural and Multidisciplinary Optimisation**
07th -12th, June 2015, Sydney Australia

# Common Automotive and Aerospace Requirements for Commercial Structural Optimization Software

**Steve Georgiadis[1], Vladimir Balabanov[2], Rodney Dreisbach[2], David Trop[2], Moritz Frenzel[3], Daniel Heiserer[3], David Keller[3], Markus Schemat[3].**

[1]The Boeing Company, VIC, Australia; steve.georgiadis@boeing.com
[2]The Boeing Company, Seattle, WA, USA; vladimir.balabanov@boeing.com
[3]BMW Group, Munich, Germany; moritz.frenzel@bmw.de `

## 1. Abstract

To widen the scope of structural optimization applications at the enterprise level in the automotive and aerospace industries, and to increase the community of engineers using integrated structural optimization software effectively, certain industrial needs have to be met by such software a priori. These needs, which are yet to be met, are discussed in this paper. Satisfying these needs includes establishing a new form of collaboration between software developers and industry and will bring applications of the integrated structural optimization software in the automotive and aerospace industries to new heights. The subject needs are not limited to commercial structural optimization software, but should be viewed as desired features in any integrated structural optimization software.

## 2. Keywords

Structural Optimization, Commercial Software, Integrated Software, Intelligent Software, Industrial Application, Computer Aided Engineering CAE, Collaboration

## 3. Introduction

Over the past decade there has been a significant growth in industrial structural optimization applications. Capabilities of Commercial Structural Optimization Software (CSOS) have increased considerably, now allowing for incorporating analysis results ranging from stiffness and strength, to crashworthiness, and durability. Some of these analyses are performed inside the CSOS. More importantly, CSOS allows for linking with the analyses performed outside of CSOS. This is a reflection of the current situation in the automotive and aerospace industries where no structural part is designed without performing a vast number of different numerical analyses. Analysts and designers need to account for an ever-increasing number of requirements to produce realistic industrial designs. The fact that structural optimization can now be applied to total vehicles with the number of influential parameters increasing from tens to thousands has only added to the complexity of the tasks to be performed by the CSOS.

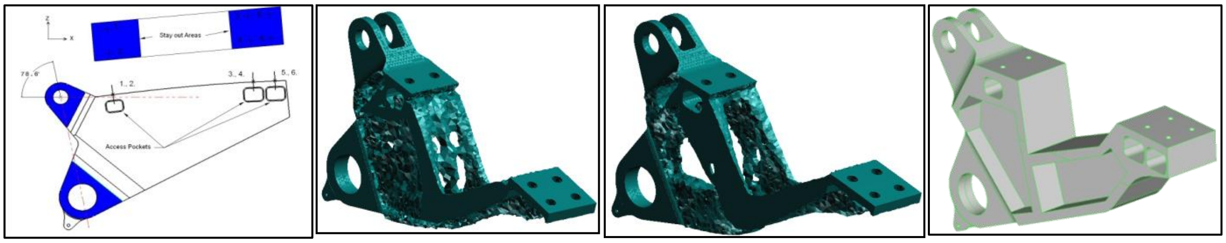
From an enterprise perspective it is often advantageous to utilize CSOS as opposed to Process Integration and Design Optimization (PIDO) tools: PIDO tools typically require their own sub-processes to be created. PIDO tools also often impose additional requirements on the analysis programs and sub-processes to be included in PIDO. Thus, the designers and analysts have to master the challenges arising from the complexity of the new enhancements and extra processes. This approach can be contrasted with using CSOS, which typically result in simpler final processes that are easier to maintain.

In spite of the analyst's efforts, even when a specific optimization process is effective, it is often not suitable for wide applications from an enterprise standpoint. One of the main reasons being the requirement to convince the approving/certification authorities that the obtained solution is a viable one.

To improve the current CSOS capabilities and to increase the community of engineers using structural optimization effectively, CSOS must adapt to the realities of the industrial design processes. Several broad categories where improvements should be made in CSOS are presented below. The listed needs are not limited to CSOS, but should be viewed as desired features in any integrated structural optimization software.

## 4. Ease of use and flexibility

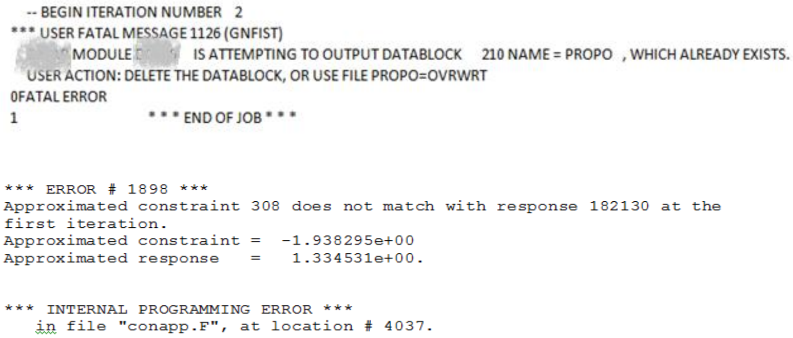
CSOS must be robust and easy to use by people with limited or no optimization background. Example of what can happen with the current CSOS usability is presented in Figure 1. Figure 1(a) shows the design space for a highly loaded fitting. The fitting was optimized using several CSOS tools. Figure **1**(b) shows two of many solutions obtained via topology optimization. All of the CSOS vendors struggled when using their own tools to arrive at a valid solution. As shown in Figure 1(c), valid solution was eventually found after an exhaustive combination of topology and shape optimizations requiring high levels of analysis and interpretation skills.



(a) Design space (b) Topology optimization possibilities (c) Valid solution

Figure 1. Stages of a typical topology optimization problem

In addition to the expertise required to operate CSOS, effective debugging also presents a problem. Two examples of error messages that have been encountered when solving structural optimization problems are presented in Figure **2**. In both cases, the software developers could not easily determine the cause of these errors and how to fix them. It is obvious that these messages are not interpretable by the most end users of the software.



```
  -- BEGIN ITERATION NUMBER   2
*** USER FATAL MESSAGE 1126 (GNFIST)
 [redacted] MODULE [redacted]  IS ATTEMPTING TO OUTPUT DATABLOCK    210 NAME = PROPO  , WHICH ALREADY EXISTS.
  USER ACTION: DELETE THE DATABLOCK, OR USE FILE PROPO=OVRWRT
 0FATAL ERROR
 1                     * * * END OF JOB * * *


*** ERROR # 1898 ***
Approximated constraint 308 does not match with response 182130 at the
first iteration.
Approximated constraint =  -1.938295e+00
Approximated response   =   1.334531e+00.


*** INTERNAL PROGRAMMING ERROR ***
   in file "conapp.F", at location # 4037.
```

Figure 2. Example of non-interpretable error messages from CSOS

Easy access to clear step-by step "How-to" tutorials on various aspects of using the specific features of the software is mandatory. These tutorials should not refer to other tutorials; they should not require additional reading to accomplish the tasks described in the tutorial. The majority of the finite element (FE) structural optimization software on the market lacks such tutorials.

CSOS must provide enough controls to expert users to access the solution details, the details of the optimization algorithms, and allow re-adjusting considerably the optimization procedural steps. For example, most of the FE CSOS has a limited control over how many times the gradients of the external response should be evaluated during a design cycle. In case of thousands of variables and computationally expensive external responses, the external finite difference gradient calculations may take a significant amount of time. Allowing users to control how many times the response gradients are calculated can significantly reduce the computational time required for optimization.

## 5. Integration with legacy tools and flexibility of internal equations

CSOS vendors often strive to bring more and more analysis capabilities inside of their software. Although helpful in some cases, such an approach is not always acceptable. Both automotive and aerospace industries have legacy tools that are required to be used, regardless of how easy to use, accurate, and effective CSOS might be. Such tools are trusted by the experienced designers. Such tools have been verified and validated as required for product certification. Intuitive and easy interfaces to such legacy tools with minimum or no programming experience are essential. The ideal CSOS should allow external tools to calculate not only the responses, but also the element and material properties, as well as variable values – everything that currently can be computed inside of Finite Element Analysis (FEA) codes. In addition, it is essential to allow for handling not only scalar inputs and outputs from external codes, as it is done now, but also arrays of inputs and outputs via a single call to external programs. Current capabilities to link to external programs often require extensive knowledge of computer systems, access to very specific versions of compilers, and having administrative privileges on the computer. Analysts don't have any of that. Struggling with linking external codes into CSOS often eliminates any desire to use structural optimization. Efficient and easy-to-use linking to external analyses is an essential part of establishing CSOS processes at the enterprise level.

To prevent calculating many characteristics outside of CSOS codes, increasing the flexibility of the internal equations could be very helpful, as well as allowing the use of some programming language syntax in the equations. Alternatively, an ability to translate from a programming language into the equations would be very useful to the analyst. The current equations provided by CSOS have similarities to the Fortran function syntax; however, these equations don't allow any loops of "if" statements. This makes them very limited and restrictive, as it requires tedious and error-prone translations of existing routines from programming languages into the CSOS equations.

It is not uncommon for the equations to reach hundreds of lines in length. Difficulties with handling the equations combined with the error-prone strict requirements for the way of compiling external analyses for CSOS often forces finite element codes to be used for analysis only. In this case FEA is coupled with PIDO optimization engines due to flexible internal equations and easier and more flexible interfaces to legacy codes.

This problem will get worse as the automotive and aerospace industries increase exploiting applications of non-metallic materials. Specifically, CSOS make use of failure criteria which are of little use for practical structural design and which are not recognized by the product certification authorities.

## 6. Intelligence and guidance

CSOS should support and guide users during all phases of the optimization process: during problem setup, execution of the optimization procedure, and when reviewing the results. The tool intelligence should be derived from the intermediate solution results and from efficient incorporation within the tool the knowledge and expertise of existing in-house tools, processes, best practices, and design criteria.

### 6.1 Intelligence and guidance during optimization problem setup

CSOS developments have already gone through great lengths of making it easier for end users to set up all aspects of a design problem in a graphical way. Often, even the order in which menu items are presented, suggests the most reasonable way for the sequence of operations to be set up for a specific optimization problem. One of the inconveniences that users experience in the current problem setup procedure is too many mouse clicks required to select various sub-options to define the problem correctly. By the time the user starts setting up the optimization problem, CSOS already has the information about the associated FE model. Thus, the choices the user makes in setting up the problem are not entirely arbitrary. An example of code-embedded intelligent guidance would be to allow the user to define the type of optimization problem first (topology, sizing, etc.), then - what specific part should be optimized, and some general parameters of the problem. Based on this information the software may be able to automatically define a majority of the parameters and relations between design variables and responses, leaving the user merely to check and adjust what was not correctly identified by the default setup.

### 6.2 Intelligence and guidance during execution of the optimization

The current CSOS approach to running optimization problems is to have the user set up the problem and then let the optimizer arrive to a better solution point without interruption. However, user intervention, user decision, and appropriate visualization technique are essential during the solution phase of complex problems. Especially when the optimizer encounters problems in finding a good solution and/or when the model and/or the objectives and constraints are not well defined. Possible techniques allowing analysts to target specific region of exploration are illustrated conceptually in Figure 3.

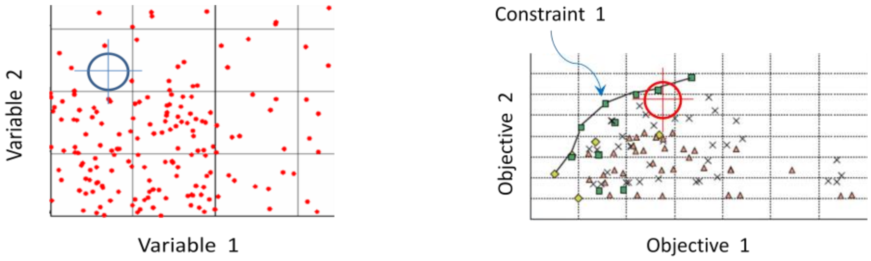


Figure 3. Conceptual examples of visually indicating the target (preferred) region of exploration where optimization should be concentrating its efforts

Note that such kind of visualization should be dynamic and accessible while an optimization procedure is being executed, thereby allowing a user to influence the direction (path) of optimizing if so desired. Using such visualization techniques allows the analyst to decide whether to tighten or to remove certain constraints on the fly during an optimization solution.

Tightly connected with flexibility of exploration is the robustness of the code. Currently, if the code encounters an error, the optimization procedure stops, asking the user to fix the error. A much more robust approach would be for the code to continue the optimization/exploration process regardless of the error. CSOS has the information about all the previous, successfully analyzed solution points; therefore nothing should prevent the software to go to one of the previously visited points, adjust the search direction, and continue the optimization, while keeping detailed textual and graphical logs of what has happened and what has caused the setback.

### 6.3 Intelligence and guidance during optimization post-processing

Visualization is essential after a candidate (optimum) solution is obtained. It is not enough to arrive to a good solution. It is essential to convince the approving and certification authorities that the obtained solution is viable. The analyst is required to clearly, visually, and without much effort, explain *why* the specific solution was obtained, *how* it was obtained, and *what happens if* some parameters of the solution were to be changed.

To answer these questions, specialized optimization-oriented visualization capabilities need to be created. Currently, most CSOS visualization tools create contour plots of responses superimposed on the structural model. These types of plots are analysis oriented. This means that for the analyst to visualize the optimization *process* he or she must go through a number of analysis result plots (e.g., stresses for several load cases), determine which ones contain critical results for the most important load cases, and then try to create the combined plot for several important iterations.

Automating and easing up this process would help to answer the “how” questions. Dynamically showing how the locations of the critical regions change from one iteration to another would also assist in managing the critical regions in the structure. Such dynamic and static contour plots on the structural model should also identify the specific load cases which caused the critical regions to appear.

Another example is the ability to immediately and easily visually compare analysis results of different iterations. All the required information for this is readily available inside the CSOS, but it is not currently presented to the analyst in a convenient form.

Readily available 2D plots of the objective function, constraint violations, and design variables with respect to the iteration number are useful from the conceptual standpoint. However such plots don't provide insight into the optimization procedure, and are not able to answer the above-mentioned "how, why, and what if" questions. To help answer the "why" questions, advanced multidimensional visualization techniques may be required.

Figure 4 illustrates an approach suggested in Ref. [1] when 2D alpha plots were used to study the tradeoffs between three airplane planform configurations. The objective values, as well as the constraints, were represented in a single plot to illustrate where the designs were located with respect to constraints. Many other multidimensional visualization techniques of this type are already available.

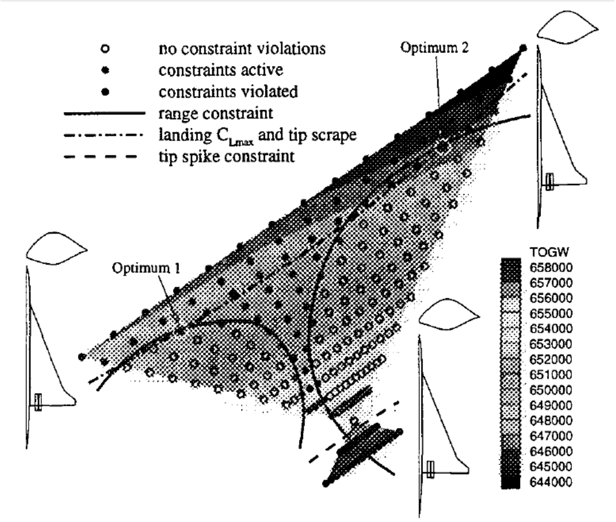


Figure 4. Illustration of a multidimensional visualization technique for a non-convex design space from Ref. [1].

Another feature desired by many analysts is the ability to visualize the response sensitivities, as well as the sensitivities of the objective and constraints superimposed on the structural components, much like the contour plots for stresses. The conceptual example of such a plot is presented in Figure 5. Such types of plots help the analyst to answer potential "what if" questions. It is even more desirable to make such plot interactive, so that the user can actually change some parameters on the plot, and observe changes in the results.

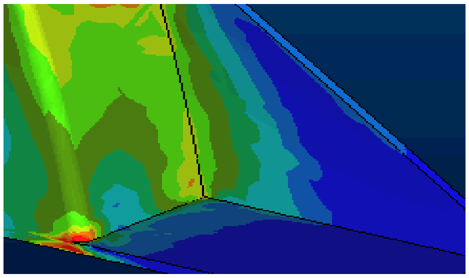


Figure 5. Conceptual example of a contour plot showing sensitivities of a structural response

Along with displaying sensitivity information comes the CSOS ability to identify the most influential variables and to suggest which variables could be eliminated or made constant. As the optimization problems are at the point when having thousands of variables is not an unusual setup, a capability to filter out important and non-important variables automatically and represent these choices graphically will be a big help to analysts.

A CSOS with enhanced intelligence may suggest users how to improve/augment design variables during (or after) optimization. For example, if a stiffness response is sensitive to thickness design variables and the loading is dominated by a bending behavior, it is reasonable to assume that introducing shape or topology variables will result in more improvements compared to the thickness variables. Thus the software may suggest possible improvements/augmentations to the existing design variables as well as the location on the part where such new variables could be defined. The software could also define these new variables automatically using default settings if a user wants that.

Extremely desirable is the ability to calculate approximate results on the fly as a response to changes in some design parameters, without performing complete FEA solutions to answer “what if” questions. Such a capability would be especially useful for large models, when FEA takes considerable amounts of time.

## 7. Optimization-related capabilities

It is common in the automotive and aerospace industries to consider dozens, hundreds, and sometimes thousands of load cases. Some of these load cases are not relevant for finding an optimal design as they don’t result in active constraints. For such cases, the CSOS should have an option to automatically and robustly handle a multitude of load cases, neglect the non-critical ones, and consider only the relevant load cases and constraints in solving the optimization problem. This procedure of load case screening should be dynamic, where the active load cases should be adjusted as the optimization process progresses.

Optimization is not the only approach to design space exploration and finding a better solution. Design of Experiments (DOE) and trade/sensitivity studies are quite popular in the aerospace and automotive industries. Currently, even if FEA must be used in such studies, the studies themselves are performed outside of CSOS with the problem setup being started from scratch. Whereas after the design optimization problem is setup in CSOS, everything is ready for both DOE and sensitivity studies. Yet these studies, which are often a part of the internal optimization algorithms, are not exposed to the user and appropriate visualization is not being created for them.

CSOS codes always proudly and rightfully claim to utilize the high quality next-generation internal approximations. CSOS could more extensively use these approximations. Such techniques as approximate Pareto optimization, global optimization, probabilistic optimization, and user-guided optimization, could all use these high quality internal approximations. The acquired data could be used by CSOS to provide analysts with a large variety of options for efficient and fast design space exploration and to arrive at a solution that is valuable from the analysts’ standpoint, and not only from the optimization algorithm’s standpoint. This is best illustrated by the fitting example in Figure 1, where the optimization algorithm continued to return a solution which was not practical for manufacturing and required the analyst to intervene several times prior to arriving at a valid design.

## 8. Collaboration between industry and CAE vendors

In the past, Computer Aided Engineering (CAE) software vendors have been co-located with industry on design projects for the wrong reason: too often, the software has been difficult to use and required vendors to be co-located with the company’s engineering team to interpret the solution results from the CSOS. An important issue in accepting CSOS in aerospace and automotive industries at the enterprise level is the early interaction between the software vendors and customers (industry) regarding the features that are planned for implementation in CSOS. Up to this point, the CAE vendors often have developed new features that the vendors thought would be important. These features were then implemented in the CSOS in a way that the vendors thought would be convenient for their customers. As a result, industry often received the newly-developed software that was not doing exactly what was needed. The specialized intermediate fixes and updates issued by vendors to bring software closer to exact industry needs were hard to install due to computer security in industry. As a result, the needed capabilities were often obtained by industry a year or two after the features were officially first released. This situation would change if software vendors were to discuss the new features (analytical requirements) with customers before starting development of the features. Not only functionality, but also appearance, performance, ease of use, and future extendibility of the new software enhancements should be discussed through ongoing technical dialogs between industry users and CAE vendors.

## 9. Conclusions

The analytical software requirements discussed in this paper would allow CSOS to gain more industrial support and a faster increase in its user-base across multiple industries. The reason for such requirements to surface at this time is that 10 years ago the CSOS were not mature enough to be seriously considered for use in the enterprise product design processes. Now, with improved CSOS capabilities and robustness, the possibility of exploiting CSOS tools more extensively in the enterprise design processes is real; however, such an opportunity comes with new and stricter requirements. Without fulfilling the proposed requirements, including advanced problem-solving techniques in addition to embedding decision-making algorithms through predictive analytics, the CSOS user-base in the aerospace and automotive industries will most likely remain static or grow slowly. This is primarily because there would be significant difficulties in persuading program management to attempt utilizing the current optimization tools more extensively in the enterprise-wide detailed-design processes. Although CSOS is the primary target of the subject needs, they are not limited to CSOS but should be viewed as desired features in any integrated structural optimization software.

## References

[1] D. L. Knill, A. A. Giunta, C. A. Baker, B. Grossman, W. H. Mason, R. T. Haftka, and L. T. Watson, “HSCT Configuration Design Using Response Surface Approximations of Supersonic Euler Aerodynamics”, 36th AIAA Aerospace Sciences Meeting and Exhibit, 1998, DOI: 10.2514/6.1998-905.