Cylinder Head Water Jacket Fatigue Optimization Workflow

Zachary Yates

Project Engineer, Caelynx LLC

***Abstract**: The water jacket is an integral part of the engine, ensuring that liquid coolant can flow efficiently over the cylinder heads. However, this structure is also very susceptible to fatigue because of the high pressure and loading frequency to which it is exposed. This paper describes a process we developed to optimize the fatigue durability of a cylinder head water jacket using multiple products from the Simulia portfolio.*

We used Abaqus to generate accurate stress and strain data, fe-safe to calculate fatigue life damage based on the Abaqus results, and Tosca to optimize fatigue performance of the water jacket internal surfaces while minimizing part mass. The large and complex nature of the part rendered a full finite element model too computationally expensive for even a single optimization loop. Therefore, in order to overcome this challenge, we used Abaqus submodeling techniques to run extensive optimization iterations in a fraction of the time.

The results of this study provided excellent design guidance for the sample part and yielded a workflow that could be applied to optimize the durability of other complex components.

***Keywords**: Automotive, Cylinder Head, Damage, Engine, Fatigue, Fatigue Life, HPC, Nonlinear Materials, Safety Factor, Submodeling, Thermal Stress, Water Jacket*

1. Introduction – Analyzing the Cylinder Head

Figure 1. Cylinder head exterior

The engine cylinder head seals off the tops of the cylinders and provides structural support against combustion cycles. These combustion cycles are high-pressure, high-temperature, high-frequency events that propagate a wide array of interrelated engineering concerns, including sealing and distortion, metal temperature and heat dissipation, and stress and fatigue. One of the aspects of cylinder head design that addresses these concerns is the water jacket, which mitigates high temperatures by channeling liquid cooling through the cylinder head. These hollow water jacket channels, while directly beneficial to temperature, adversely affect performance against combustion pressures. This makes it important to have the right water jacket design that will do the most good for temperature and the least harm for pressure.

It is implicit that we will use Abaqus to analyze these stress conditions. However, combustion cycles transpire with high frequency, and while Abaqus can analyze the effects of a single loading event, we will need fe-safe to analyze the cumulative effects of that load in repetition. Beyond that, we need to consider how, as engineers, to process and act on the analysis results. While we could, in traditional fashion, interpret the results, redesign, and reanalyze the part ourselves, we could also utilize additional products in the Simulia portfolio to automate that process to a significant degree. With these additional Simulia tools, the process is called optimization, and requires the following:

1. a clearly defined scenario (i.e. a loading and constraint condition)
2. metrics by which to judge performance (e.g. stress performance in a fillet, first natural frequency, part mass)
3. modifiable parameters that affect performance (e.g. part features whose size or thickness isn't fixed by other concerns, empty voids that could potentially be filled with material)
4. the compute time and power to incrementally change the aforementioned parameters until the desired improvement in the metrics is realized

While it will not typically provide the exact design solution, optimization can provide clear and specific direction that has been holistically developed through iterative, probative analysis, unlike a precisely targeted conceptual solution that may create unintended consequences once put into practice.

Developing an optimization procedure is a multistep process which presents its own unique concepts and challenges. Understanding these concepts and solving these challenges gives us a roadmap from static analysis to optimized design that makes novel use of Abaqus capabilities at a large scale. This methodology, created for something as complex as a water jacket cylinder head, is easily adaptable to similar analyses on any scale.

2. Simulia Workflow

A large scale complex optimization within the Simulia portfolio relies largely on Tosca, a powerful tool with which one can automatically balance multiple design restrictions to achieve structural goals of one's choice. For example, one can aim to minimize part mass while keeping fillet stresses under 50% yield. In order to run, Tosca requires, at minimum, a solver input file and an optimization parameter file. However, Tosca can operate on fatigue results as well, simply by providing a fatigue deck to follow the Abaqus deck. The workflow for a cylinder head water jacket fatigue analysis in Tosca is simple: Abaqus for stress analysis, followed by fe-safe for fatigue calculation, and then Tosca to iterate on the Abaqus/fe-safe cycle. Development of the Abaqus and fe-safe analyses can proceed as normal, but coming up with the ideal Tosca analysis is a bit different. It will involve making some decisions, possibly through trial and error, as to which parameters to optimize for, which parameters to use as boundaries, exactly the size and location of optimizable surfaces and volumes, and any limits to the alterations Tosca shall perform.

3. Building the Models

The first step in a fatigue analysis is to create a model that provides fe-safe with the data it requires, i.e. stress/strain output from a static analysis (Fig. 2). Working backward from there, that means high quality, detailed mesh capturing all geometric features. That also means an analysis procedure that doesn't sacrifice the accuracy of its interactions with abstractions and approximations for the sake of run time efficiency. A typical high-quality cylinder head water jacket analysis has a relatively lengthy load step sequence:

1. Valve seat press-fit
2. Bolt clamping
3. Temperature mapping
4. Multiple firing loads

In the case of our cylinder head, the FE model comes in at 5,381,202 nodes, 3,791,985 elements, and 10 load steps, featuring castings, liners, gaskets, valve seats, temperature-dependent, elastic-plastic materials, and contact between all parts. On a 60-CPU HPC cluster, it takes 76 hours to run. Between Abaqus, fe-safe, and Tosca, Abaqus is likely to be the most computationally expensive, and one can expect it to be the primary practical concern in managing an optimization project. A cylinder head water jacket is a great test case for developing an optimization methodology precisely because this issue is so magnified – repeated iterations of large implicit analyses will rapidly eat up both computation time and data storage.

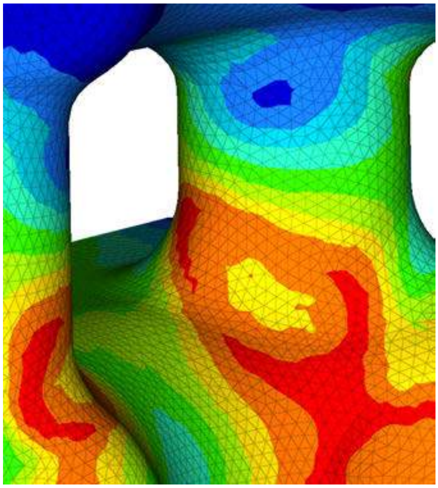
Figure 2. High-quality water jacket mesh with stress contours

The second step to a fatigue analysis is simply to run the fatigue calculations, which are far less demanding on resources than the static analysis. The type of firing cycle here – firing loads and unloading – is perfect for fe-safe's load cycle capabilities, which process a series of transient stress patterns rather than simple mean and amplitude stress. With stress gradient, surface roughness, temperature dependency, and statistical influence; we've used just a few of the abilities in fe-safe's repertoire, to say nothing of other unused capabilities like stress rearrangement, boundary layers, and surface residual stresses. The results of the fe-safe run (seen in Fig. 3), while short in calculation time compared to Abaqus, may present their own challenges. In the case of this water jacket, that challenge is the number of fatigue hotspots that exceed the acceptable damage threshold. Each hotspot increases the complexity of the problem, and may have implications for Tosca runtime down the line.

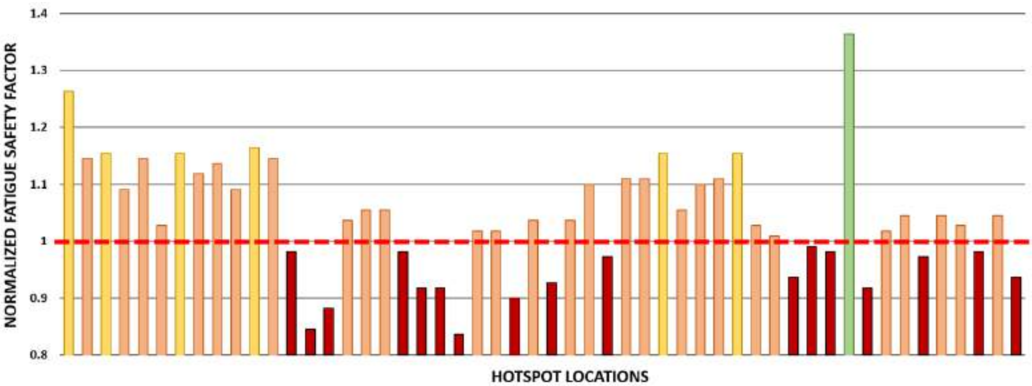

Figure 3. Noted fatigue hotspots on cylinder head water jacket

After defining the fatigue analysis comes defining the optimization, via the Tosca parameter file, which runs the Abaqus and fe-safe inputs, tells Tosca how to process the results, and outputs a new model for Abaqus and fe-safe to run again. Arriving at the ideal parameters to use here constitutes the bulk of development time. The first decision of a Tosca analysis is which type of optimization to do. Topology optimization creates an entirely new design in a given volume of space, while shape optimization introduces smaller improvements to an existing design. This water jacket fatigue optimization is the latter type, as the general design is already complete and we are interested in local improvements.

For our shape optimization, we provide Tosca with a surface area whose fatigue damage it is to monitor, and a (probably larger) area whose shape it is free to manipulate to improve those fatigue calculations. The bounds of that manipulation, i.e. how far Tosca and grow or shrink a surface, and the size and shape of the surface itself, are one of the key things to investigate when designing a Tosca analysis. One must also come to understand whether it is better in any given case for Tosca to concern itself with the average or the maximum damage over the surface in question. This choice can set Tosca on differing paths toward an optimized solution. One may also need to account for design constraints such as constraining shape manipulations around a bore to a specific axis. It may also be wise to engage Tosca's mesh smoothing capabilities to keep the element quality sufficient for Abaqus.

4. Fitting for Tosca

We know that the Abaqus model takes very long to run and that the fe-safe hotspots are numerous, and these two issues combine to present time resource challenges to running Tosca optimization. The direct solution is to bring the numbers down, i.e. run a smaller Abaqus model and address fewer hotspots. These can both be done at the same time.

A cylinder head is a repetitive part, so if we can improve one area, we can likely apply the same changes to three, four, or more areas. Fig. 4 shows two examples of symmetry that would result in redundant analysis. While thermal loading may not be symmetrical when the geometry appears to be, taking the worst-case scenario while keeping this point in mind is a viable compromise. Just this principle alone reduces the scope of the work for this particular water jacket from fifty-two optimizable locations down to eighteen. Not only does this reduce the number of optimization tasks to do, but it also reduces the complexity of the analyses that remain. The fewer variable relationships we have to manage and compute, the better.

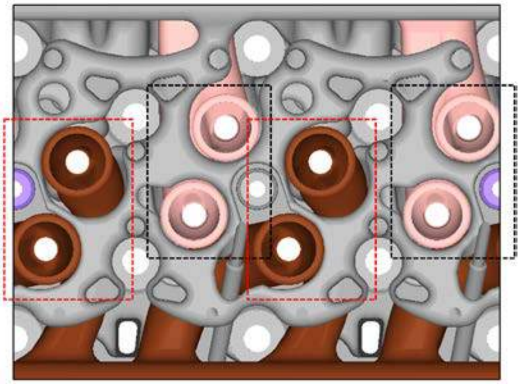

Figure 4. Water jacket symmetry

Abaqus has a feature that is specifically made to address challenges of run time. The submodeling technique allows the analyst to run a relatively coarse model first, then use those results to drive the same analysis on a smaller location within the global model that has been meshed in greater detail. In Fig. 5, you will see an example of how a submodel region might be cut out from a global model. This idea of focused computing effort, while originally intended for conventionally analyzing an individual load case, still directly applies to our optimization, i.e. improving known hotspots within a formidably large model. While a regular stress analysis requires that better mesh detail, so does Tosca – it needs good mesh with consistent edge lengths so that it can manipulate them evenly and precisely. While a complex input deck may require a lot of run time, that time constraint is only compounded further by iterative optimization.

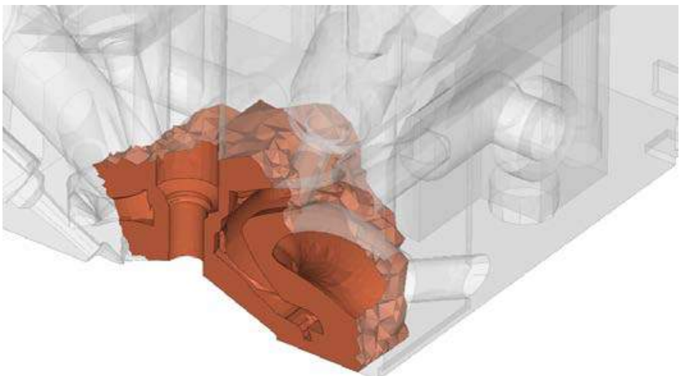

Figure 5. A submodel region

Our submodels must include the eighteen hotspots which we narrowed down from fifty-two according to symmetry. However, that does not mean we need eighteen submodels. When those hotspots are close enough to influence each other, a submodel should encompass the both of them. In the end, for our water jacket optimization, that means eight submodels. These submodels are zones of influence – portions of the global model – across which one might expect a given hotspot's optimization to have a structural effect. With each of those submodels completing in less than an hour, and with the ability to run multiple submodel optimizations concurrently on less powerful machines, we can execute a whole round of targeted optimizations, start to finish, in the time it would take to do one global model iteration.

5. Results

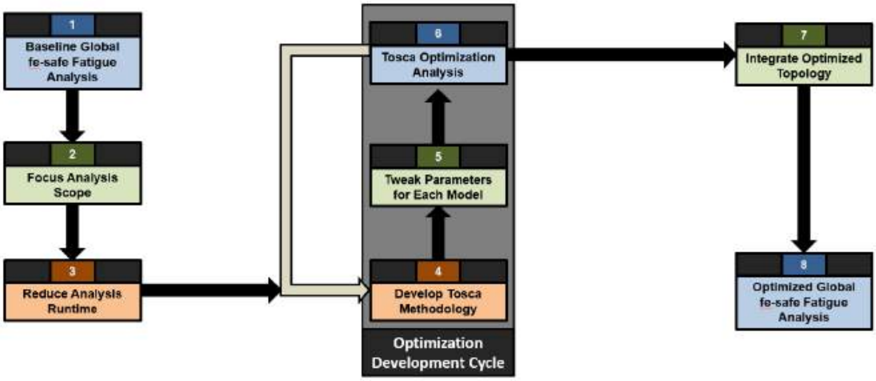

Figure 6. Full optimization workflow

The entire process consisted of eight major workflow steps, as seen in Fig. 6 above, with most of the time spent cycling steps 4 through 6, refining the submodel optimizations to get the very best results. Changes could be as simple as redefining Tosca parameters such as displacement and volume limits, objective targets, output scaling, or spatial constraints. Occasionally, some submodel selections had to be changed, for example, when it was clear Tosca wanted more water jacket real estate to make its changes. The optimized mesh was then integrated back into the global model and the full seventy-six-hour global fatigue analysis was run again to verify the improvements observed in the submodels.

Figure 7. Increases in safety factor after optimization

We succeeded in developing a highly-structured workflow that integrated Abaqus, fe-safe, and Tosca into an effective fatigue optimization process, as one can see in the safety factor improvements in Fig. 7. Some of the biggest safety factor gains involved the emergence of new gussets where before there were none. Several fillets were broadened, bores lengthened, channels widened, and some vertical pillars even shifted orientation (to great effect), as visualized in Figs. 8 & 9.

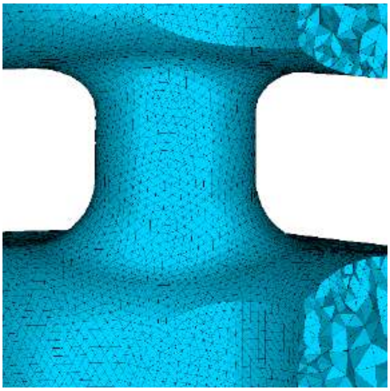

Figure 8. Pillar before optimization

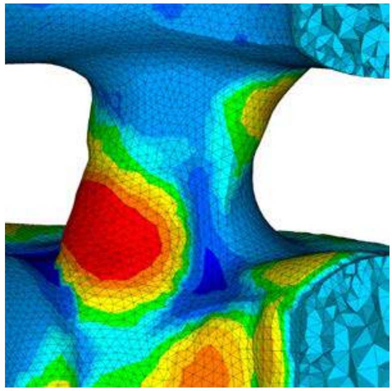

Figure 9. Pillar after optimization, plotting optimized surface displacement

Abaqus provided the necessary easy-to-use tools to control a very large and detailed model; fe-safe's fast and powerful code provided the fatigue output required, utilizing several complex inputs and influence factors; and Tosca worked seamlessly with the Abaqus and fe-safe solvers to produce substantial, verifiable results on a comfortable time scale. With a little understanding and ingenuity, Abaqus, fe-safe, and Tosca put great results within our grasp. If you'd like to learn more about this developed methodology, please contact Caelynx for more details.