CHAPTER 15
Structural optimization

15.1 Introduction

Even in the field of structural optimization, since the first calculation programs started to make available dedicated tools, the steps taken have been enormous. It is clear that the ambitious goal of optimizing a structure is very attractive and certainly interests the designer of a given product as much as the final user of the product itself.

At present we can basically identify three categories of structural optimization.

15.1.1 Size optimization

Size optimization allows to obtain the optimal values for certain parameters describing a given structure, such as material properties, shell element thicknesses or the characteristic dimensions of beam element sections. This is historically the type for which the first steps were taken. It must be said that, if it were possible to express all the characteristic quantities in analytical form, the achievement of the optimal configuration could be easily obtained through the techniques of Nonlinear Programming (NLP), with which it is possible to calculate the constrained minimum (or maximum) of a function of several variables. Clearly such an approach cannot be adopted in a general sense, as stresses and deformations are calculated numerically by a FEM code. The methodologies used in this case are the most varied and we will not dwell on them. It is sufficient to know that with this type of structural optimization it is possible, for example, to minimize the mass of the component while respecting certain geometrical parameters and reaching a certain stiffness objective. Such codes are generally able to optimize the thicknesses of shell elements, the section properties for beam elements, find the best material and even vary the orientation angles of the sheets and their stacking sequence (see Chapter 8) in the case of structures made of composite material. From what has been said it is evident that the structures which can be optimized with this methodology must be modeled with beam or shell elements. In any case, the structure to be optimized is already defined in its fundamental characteristics (overall geometry).

15.1.2 Shape optimization

Shape optimization is a methodology for optimizing the weight and behavior of structures by applying variations to their geometry. For example, local stress peaks can be reduced by locally modifying the curvatures. Also in this case the structure is already defined in its globality. The difference with the previous case is related to the fact that the optimization code can operate, even if not dramatically, on the geometry of the part, moving the nodes of the model (automatic morphing of the elements). In this case, the structures to be optimized can also be modeled with solid elements. With the advent of parametric pre-processors able to manage, almost like CAD tools, geometric changes and to automatically reshape the new model, shape optimization can be successfully applied even when substantial changes to the morphology of the component are needed. Of course, if the geometry to be used could also be expressed (as well as the constraints and objectives) in analytical form, NLP techniques could also be successfully used for shape optimization; indeed, to be picky, before the advent of topological optimization no distinction was made between shape and size optimization.

15.1.3 Topological optimization

Topological optimization represents the ultimate frontier in structure optimization. The methodology can, indeed must, be applied from the very early stages of the project. In fact, for the code to work, it is necessary to define a volume within which the final part will be enclosed. This space is clearly limited by the dimensions and functionality of the objects which are next to the body under examination.

Theoretically, it would be sufficient to define this volume, the material with which you would like to make the part, the constraint and load points, the forces, the design variables, the constraints of these variables and the objective, and press a key to obtain the structure optimized for that specific function (the load conditions can also be more than one). The basic principle is quite simple: the code starts from the full volume (which must clearly be meshed with solid elements) and applies loads and constraints, performs the calculation, evaluates the strain energy it obtains on the volume and assigns to the elements with low strain energy for the next iteration a very low "stiffness density" value; in this way, without deleting the elements from the model, they will contribute in a negligible way to the stiffness of the structure at the next iteration. It is thus possible that, during the optimization process, some elements "die" (low density) and then "resurrect" (normal density). When the calculation has reached convergence, it is possible to plot the elements above a certain density threshold: this will be our optimal structure for that or those load conditions.

From the practical point of view things are clearly not so simple; for example it could be necessary that some areas of the volume remain unchanged because they are wanted in a certain way; or it could be necessary not to have material below a certain thickness (for example for technological reasons); or it could happen that the code creates shapes that cannot be obtained in any way. All these constraints can clearly be passed on to the computational program, but sometimes unexpected results can occur. The combination of topological optimization and 3D printing has recently opened new horizons, since with the DMLS technique it is also possible to create geometries that contain "closed voids", giving greater freedom to the topological optimizers.

It must then be said that, once the optimal shape has been obtained, it must be passed to a CAD program for its engineerization; and this can present some difficulties linked to the fact that the optimized model is made up of finite elements and will generally have a "segmented" trend that will have to be correctly interpreted by the designer. In this Chapter we try to illustrate through an example how we should proceed in a project involving structural optimization; we will do this without going into too much detail, but trying to give some guidelines.

15.2 A case study

We want to design the chassis for a single-seater racing car that has the maximum torsional stiffness compatible with a given mass limit (for example determined through a study of vehicle dynamics), a volume identified in part by the regulations and in part by aerodynamic calculations (CFD) and constraints to be met (such as the attachment points of the suspension and engine). All geometric constraints identify the so-called Design Space, i.e. the volume within which the structure can develop. For our simplified example we have the situation in figure 15.1, where clearly the internal recess (generally regulated) will serve for the pilot, while the external dimensions are dictated by the aerodynamic surfaces.

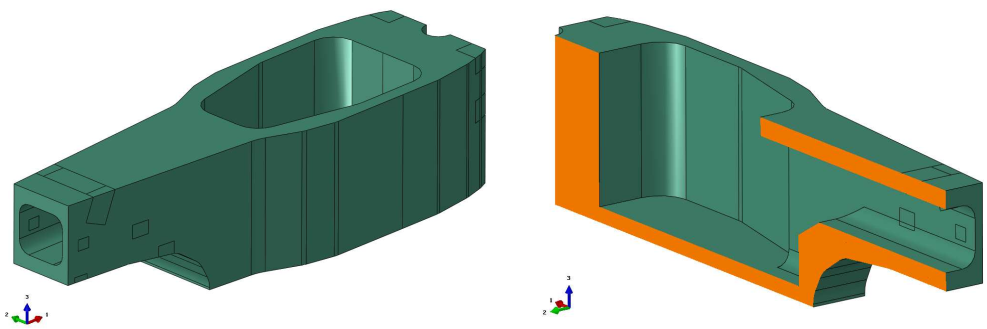

Figure 15.1. Design Space for a hypothetical single-seater racing chassis. Having imposed aluminum as a material, the mass of the Design Space is 755 kg.

Since we want to avoid any prejudice, we must be ready to implement "exotic" solutions, not even excluding a milled structure from solid, since we start from an almost solid. The only thing we feel to impose is the material with which we decide to fill our Design Space: since we want to maximize the stiffness and have a mass below a certain target, we choose the material with the best Young's modulus / density ratio, namely aluminum (we will return later on composite materials). We will proceed as follows:

1. the areas that cannot be "touched" by the optimization are identified;
2. the model is meshed with solid elements (we use 4-node tetrahedra, the real stiffness doesn't matter at this stage, see Chapter 6); the mesh must be quite refined to get good results;
3. constraints and loads are imposed (conditions may be more than one);
4. the optimization run is executed, which will require several iterations of calculations (which can be both linear and nonlinear);