CHAPTER 2
Modeling of boundary conditions

2.1 Introduction

Once the finite element model is built, it is necessary to impose the appropriate constraint and loading conditions; the boundary conditions tell the calculation program the points where the external forces are applied and where they are reacted.

The application of the constraints is generally more " tricky" than the application of the loads: in fact a model constrained in a wrong way can make the solution of the system of equations impossible. It is necessary, as we will see shortly, to ensure that the distribution of constraints is such that the structure is not labile; this means that all acts of rigid motion must be prevented, both for the complete model, and for a part of the model with respect to another (elimination of internal mechanisms).

Furthermore, the various element types we discussed in Chapter 1 have different degrees of freedom (DOF) at their nodes: for example the 3D brick element has only 3 translational DOF, while the shell element has 6, having also 3 rotational DOF. Therefore, strictly speaking, in a brick model it would be necessary to block all rotational DOF; however, many calculation codes impose constraints automatically when they "read" that certain nodes have elements without stiffness in some DOF. We will return to this aspect in Chapter 6.

2.2 Constraint conditions

We have said that a correct application of the constraint conditions must first of all ensure that the structure is not labile: this means that any type of mechanism must be eliminated, otherwise the resulting stiffness matrix is non positive defined and therefore the problem cannot be solved numerically. A fairly classic example is represented by solid element models. If, for example, one constrains all the nodes of an edge in the six degrees of freedom, one can be convinced of having created a clamp, but this is not so; in fact, as we mentioned above, solid elements do not have rotational stiffnesses and therefore are not able to react with concentrated moments. The result is a cylindrical hinge, with consequent instability. Generally this type of error is not very dangerous because the calculation codes, if they don't come out with an error message, at least warn the user that arbitrary stiffnesses have been added to make the matrix positive defined and to continue then in the solution of the matrix equation. It will be up to the structural engineer to understand what has generated the problem.

This should also make us understand that finite element models must always be constrained (exceptions are the calculation of eigenfrequencies in free-free conditions - Chapter 4 - and dynamic calculations - Chapter 14); it does not matter if we apply a balanced system of forces, because the calculation program has no way of "knowing", until it has solved the system of equations. On the contrary, the application of a system of self-equilibrated forces (i.e. actions and constraint reactions previously calculated with another method) can be a way to verify the success of the calculation, checking that on the appropriate distribution of constraints (which in this case must be rigorously statically determinate) the values of the reactions are null.

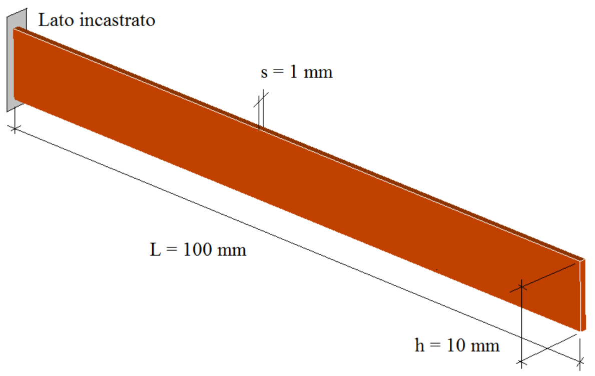

Figure 2.1. This beam is made of steel. At one end it is constrained with a clamp, while at the other end it is free.

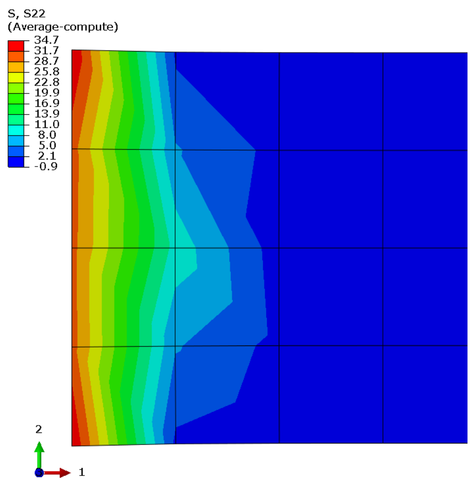

Figure 2.2. Vertical stress distribution: the maximum value of this quantity is 34.7 MPa and corresponds to about 30% of the longitudinal stress (100 MPa). 2D plane stress elements were used for this model.

Another error may be caused by the improper extension of De Saint Venant's theory valida for one-dimensional bodies; let's take the beam in figure 2.1 and stress it with a tensile force F = 1000 N.

Solid Mechanics tells us that the stress in the vertical direction, being a uniaxial stress state, should be zero in all points of the rod. However, if we observe figure 2.2, we realize that, in proximity of the constraint, such value is far from zero, while more we approach the other extremity and more effectively the transversal solicitation tends to be null. What has happened? In this case the constraints that have been given to the edge nodes prevent transverse contraction (as can be seen from the deformation, amplified by 200 times) and consequently give rise to the stress that can be seen in figure 2.2. In fact, if we now take the same model and, with the appropriate constraint conditions, we allow all the nodes of the vertical edge (except one to eliminate lability!) to slide in the vertical direction we get the vertical stress contour shown in figure 2.3. Notice how now, apart from the numerical "dirt", we actually have a null value for this stress component. It must be underlined that the most correct case is the first, because in reality the second distribution of constraints is difficult to obtain, while a weld, for example, is well represented by the first condition.

It could be objected that, since the load is applied along the beam axis, it should not be necessary to constrain even one of the transverse DOF anyway; similarly to what has been said about the system of self-equilibrated forces, also in this case the code should in any case invert a non positive defined matrix and this is not possible. We therefore reiterate that, at a minimum, it is always necessary to apply a non labile statically determinate constraint system.

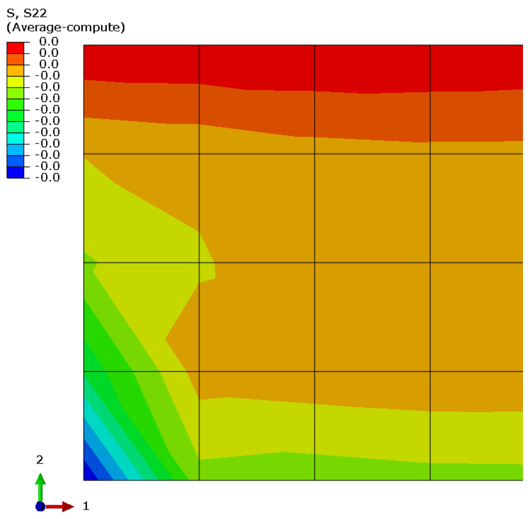

Figure 2.3. Contour of vertical stress when the distribution of constraints was changed. Now the value along the entire domain is engineeringwise null.

Perhaps it is unnecessary to point out that with a beam-element model this effect related to non-zero transverse stress in proximity to the constraint does not occur.

All this to highlight the fact that, beyond the possible introduction of lability within the model, to pay attention in the application of constraints also means thinking about the effect that these can have on the structure, on its deformations and on its stress state; even if, generally, in the points of discontinuity one tends not to take as valid the stress values produced by a finite element model.

What just stated is also valid because very frequently the constraints are applied to the central node of MPC elements, both to represent "sensible" constraints, and because the extraction of the constraint reaction, being concentrated in a single point, is immediate; for example, if MPCs were not used to constrain the suspension arm shown in figure 2.4, it would be on the one hand impossible to simulate the spherical hinge type constraint and on the other hand it would be more complex to extract the constraint reaction.

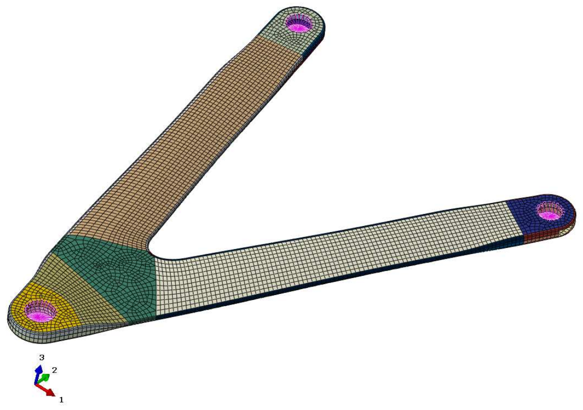

Figure 2.4. Suspension arm for racing car: MPC spiders (fuchsia in the picture) are used to apply point constraints (spherical hinges) and concentrated loads.

It is clear that not only is it essential to apply formally correct constraint conditions, but it is also necessary to ensure that what is being modeled has a physical match; for example, if an organ is constrained by bolts, the modeling of the constraint must allow free rotation around the screw axis.

It should also be said that generally the calculation codes, upon request, provide the values of the constrained reactions in the restrained DOF; this information is very precious, both for the calculation of connections, as mentioned in Chapter 1, and to verify the correct application of the loads: in fact, the codes usually also provide the resultant of the constraint reactions which, together with the resultant of the applied loads, allows precisely this control. Not only that, but if the two results are different (clearly, besides slight numerical differences) it means that some force or moment may have been "lost" due to numerical errors; we will return to this subject in Chapter 9, when we will discuss model validation methods.

One last aspect that we briefly mention concerns unilateral constraints: a typical example are supports. Unfortunately, there is not a "simple" way to realize unilateral constraints; in fact, the constraint conditions are limited to make one or more DOF somehow "inactive", while the unilateral constraint must keep active or not the DOF, to which it is applied, depending on the direction in which it is moving. And this requires iterative techniques typical of nonlinear calculations.

2.3 Load conditions

2.3.1 Point loads

The loading conditions represent the dual case of the constraint conditions; in fact, where the displacements (constrained nodes) are known, the forces (constraint reactions) are unknown, while where the forces (loaded nodes) are known, the displacements are unknown. With regard to the application of loads, it is practically impossible to create situations that can generate numerical errors.The only errors that can be made are conceptual ones; for example, as in the case of constraints, applying a concentrated moment in a node that belongs to solid elements is wrong (and it is not said that the calculation code produces a warning message) because of what has been said above.

Another example that produces seemingly unexpected results is the application of a force divided by the number of nodes that make up the segment affected by the loading condition. Let's take again the beam of figure 2.1 and load it with an axial force F = 1000 N. The vertical section is divided into four elements (and therefore five nodes); the first thing that comes to mind is to apply a force f = 200 N to each node. Figure 2.5 shows the deformation of the beam in the zone of application of the load; it is possible to observe how the nodes at the intrados and extrados deform more than the internal ones. This happens because in these positions there is only one element on which the nodal force is applied, while for the intermediate nodes there are two elements involved. To overcome this problem it is sufficient to apply half of the force that pertains to the central nodes, making sure that the total sum is still equal to 1000 N. In this case we will have:

3 central nodes = 250 N each
2 vertex nodes = 125 N each