What if the calculation fails?

The first things to do in case of a failure of the calculation are the following:

- Browse the log file (“listing”) and the documentation (user manual).
- Check that the version of the code is the expected one and that it is effectively accessible on the machine.
- Simplify the case: remove as many user-defined functions as possible to better identify the source of the failure (for example, the simulation should be checked with constant physical properties before applying variable values).
- If the user is not perfectly familiar with the computer, the case should be transferred to a machine that the user is more familiar with so as to avoid any cause associated with the operating system or with the installation of the code.
- If the simulation fails after the log file of the kernel of the code has been produced, increase the level of verbosity (IWARNI) to try and better identify the source of the problem.

If the problem persists, one may try and obtain a first converged result that could *potentially be improved later,* with the following parameters and choices:

- One should check that the initial state is “reasonable”: it is not essential to prescribe an initial solution close to the solution that is searched for, however an unphysical initial estimate of the solution may lead to a failure of the computation (for example: initial velocity oriented in the opposite direction, initial turbulent variables incoherent between each other, too large initial density gradients…). Some difficulties can be associated with the choice of the initial state (they are normally characterized by a transient that can be evacuated through the outlet of the domain by setting a time step value corresponding to a large Courant number during the first few time steps (for example: the standard value of the time step may be multiplied by a factor of 10).
- If one suspects a difficulty associated with the mesh quality, one may use the gradient computation based on an extended stencil restricted to risky regions (IMRGRA=3). For tetrahedral grids the gradient computation based on a fully extended stencil (IMRGRA=2) is the most reliable approach.
- If one observes oscillations on the advected variables, it is advised to modify the convective scheme as indicated in Card_BPG_4 (in the worst, most unstable case, a first-order upwind scheme should be used for all the advected variables, BLENCV=0.0).
- If one observes that the pressure iterative solver takes a long time to converge or that overshoots of the velocity occur locally where the quality of the mesh may be questionable, it is advised to under-relax the pressure increments (RELAXV(IPR)=0.7).
- If one observes that the convergence is difficult on a low quality mesh, the stability of a RANS computation with a turbulent viscosity model (k-epsilon et k-omega) may be improved by switching off the flux reconstruction for the variables for the turbulence – k, epsilon, omega – (IRCFLU(IVAR)=0).
- If spurious velocities appear close to a non-vertical wall for a computation with gravity and variable density, it is advisable to use the extrapolation of the pressure gradient at the wall (EXTRAG(IPR)=1.0).
- If spurious velocities appear in a region where the pressure gradient is discontinuous because of the presence of a stratification or of a head loss zone, it is advisable to use the option IPHYDR (and to take care that the parameter ICALHY is properly set when there is a head loss on cells adjacent to the outlet of the domain). When this parameter is activated, it is useless to extrapolate the pressure gradient at the wall with EXTRAG(IPR).
- If spurious oscillations appear at the outlet, it is advisable to use a Dirichlet condition for pressure to impose a flat profile, as long as this hypothesis is physically sound.

The previous analysis may lead to the conclusion that it is necessary to modify the mesh.

An example of a “safe mode” set of parameters is provided hereafter: with this choice, the priority is given to robustness and low CPU cost over accuracy:

- Turbulence model: k-epsilon with linear production
- Convective scheme: SOLU for the velocity, UPWIND for the turbulence and the scalars
- No flux reconstruction for the turbulence (IRCFLU=0 for k and epsilon)
- Threshold for the convergence of the iterative solvers: 10^{-5}
- Pressure increments under-relaxation (RELAXV(IPR)=0.7)