

Automatic Limited Compressibility

Flow Science, Inc.
December 14, 2000

DISCUSSION

Pressure-velocity iteration convergence can sometimes be difficult to achieve. Usually, situations where this is the case are described as “stiff,” which is descriptive but not of much help. In this report we present a scheme for automating the use of limited compressibility in *FLOW-3D*[®] to help reduce the stiffness of problems in which convergence is difficult to achieve.

In order to see why some situations are stiff it is necessary to first gain a general understanding of how the pressure iteration process works. The pressure iterations are analogous to a time-marching solution in which the iterations correspond to time steps. Convergence of the iterative process is equivalent to reaching a time asymptotic solution. Most iterative methods can only propagate pressure changes a distance of one grid cell per iteration because the formula for a pressure change is usually based on neighboring cell interactions. Thus, in N iterations for example, pressure signals can propagate at most across N grid cells. In many flow problems this is enough since pressure adjustments are often restricted to local neighborhoods.

In some cases, however, this is not the case. A simple example where a non-local adjustment is required can be found in the case of a long tube filled with an incompressible fluid. If the velocity at one end of the tube is changed, then the velocity must change along the entire tube to insure incompressibility. If the tube is more than N grid cells long then more than N iterations must be carried out to achieve an incompressible flow field in the entire tube. A corollary of this is that the number of iterations to get convergence will increase as the grid resolution in the tube is increased.

From a fluid dynamic standpoint, no fluid is completely incompressible. Density changes are proportional to the square of the ratio of fluid speed to sound speed in the fluid. For practical purposes, a fluid can generally be treated as “incompressible” when its density changes by less than 1%. This condition corresponds to fluid speeds remaining less than about 0.1 of the speed of sound in the fluid. In most situations the speed of sound in a liquid is very much larger than the fluid speed, but this observation about density changes suggests that the fluid sound speed could be reduced without seriously affecting the incompressible behavior of the fluid.

Using this observation about sound speeds we have constructed an algorithm to improve pressure-iteration convergence by automatically introducing some compressibility to a fluid through the quantity r_{csql} , which is equal to $1/(\rho c^2)$ where c is the speed of sound in the fluid.

The approach we have used can be understood from the following argument. Suppose we want to limit pressure iterations to a maximum of 50 per time step. This means that sound waves must be set to propagate no more than 50 grid cells in a single time step. On the other hand, numerical stability (and accuracy) conditions limit the distance a fluid particle can go in one time step to about one half of a grid cell width. Thus, the effective ratio of fluid speed to sound speed in this

case would be less than 0.01, or an order of magnitude smaller than that needed to keep density changes less than 1%.

In practice, this estimate is overly optimistic because sound waves must pass across a fluid element several times to generate a mean flow. In the tube example used above, for instance, the tube behaves like an organ pipe with one closed end and requires pressure waves to make four transits of the tube to complete one full pressure oscillation cycle. This means that rather than 50 iterations, it may take four times that, or 200 iterations, to establish the correct pressure field. If we adopt this factor of 4 as typical, we can think of the effective sound speed as being four times smaller and the corresponding ratio of fluid speed to sound speed increasing to 0.04, which implies possible density changes on the order of 0.16%. This is still a very small change and the fluid remains nearly incompressible.

The new automatic compressibility feature, which is based on the above idea, is designed to introduce limited compressibility through the variable *rssql*. The algorithm keeps the number of iterations required for convergence down while at the same time insuring that density variations remain small enough for the fluid to be considered essentially incompressible. The basic algorithm has also been supplemented with additional logic to allow for the limited compressibility to slowly float up and/or down in response to how hard/easy the solver is working to obtain a solution.

Our approach is designed to push the level of compressibility to the smallest possible value without increasing the pressure iteration count. In practice, we have found it a good idea to limit the minimum value of *rssql* to be one hundredth of its initial value, otherwise there may be some convergence problems arising from operating with excessively small numbers.

The new algorithm for automatic limited compressibility can be activated in one of two ways. The first is to set *imp=2* in Namelist XPUT. With this setting the program will introduce the automatic feature immediately, i.e., beginning with the first cycle of computation. It is best to not input a value of *rssql* in Namelist PROPS so that the program can compute what it thinks will be an optimum value.

If a non-zero value of *rssql* is defined in the input, then the automatic feature will use this value but will still attempt to adjust it up and/or down as needed. The problem with this approach is that if a poor value for *rssql* is input the program may not be able to adjust it fast enough to be effective. On the other hand, if the value chosen by the program is not good, then being able to input a better initial value is a good feature.

A second way to activate the automatic limited compressibility is to use the *ihelp=2* option, which tells the MENTOR system in the program to turn on automatic compressibility if it detects difficulties with pressure convergence.

In any case, the summary output file (hd3out.dat) contains the value of *rssql* used by the program at the time of every short print.

QUALIFYING COMMENT

Introducing limited compressibility will make it easier to achieve pressure iteration convergence, but it also introduces some compressibility effects that should be considered before using this technique. For example, in the long tube problem discussed above, it was observed that a uniform velocity is difficult to establish in the tube by iterations along, especially when the number of grid elements defining the length of the tube increases. With limited compressibility we can limit the number of iterations used in a single time step, but the consequences of this are pressure waves and velocity oscillations traveling through the tube. The oscillations occur as variations about the desired average values of velocity and pressure at steady conditions. Steady conditions are approached with a smoothly decaying oscillation amplitude, but the time interval required to reach steady conditions is approximately equal to the time it takes a fluid particle to traverse the length of the tube.

TESTING

The automatic limited compressibility feature has been successfully tested on three completely different types of flow problems. One of these was for steady flow in a long tube consisting of several straight sections connected at right angle-junctions, Fig.1a. Flow enters at the bottom through a fixed velocity boundary and exits at the right side of the plot at a fixed pressure boundary. Initially, there is no flow in the tube, so that the applied velocity at the inlet must be driven through the entire tube. Without limited compressibility the SOR pressure-velocity iteration method will not converge for many cycles. When the new scheme is used, $imp=2$, the program computes an initial value of $rscql=3.24e-6$ and no iteration convergence failures occur. By the end of the calculation $rscql$ is two orders of magnitude smaller at a value of $3.5e-8$.

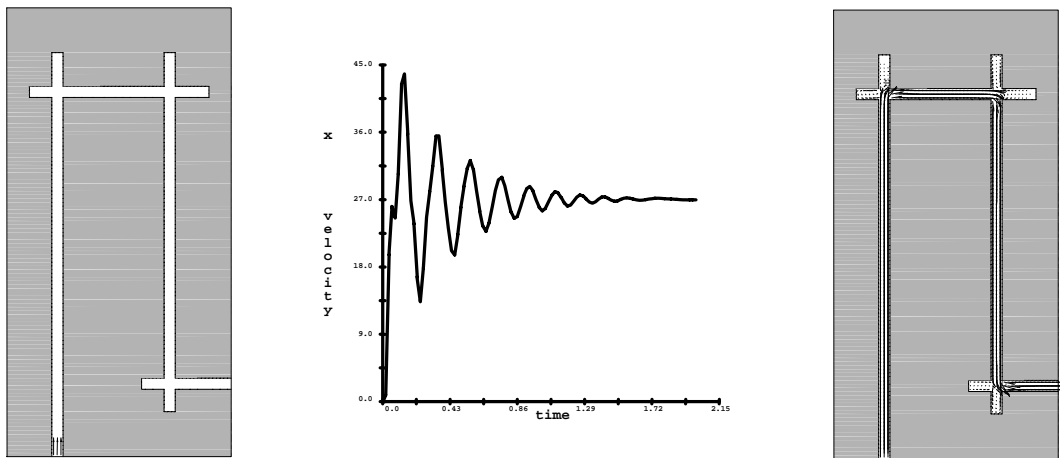


Figure 1. Tube test problem. (a) Initial setup; (b) Time history of velocity in top horizontal section; (c) Final solution.

Fig.1b shows the computed history of the velocity in the horizontal section at the top of the flow channel. This plot clearly shows the oscillatory approach of the velocity to its asymptotic value because of the presence of some compressibility. Fig.1c is the converged, steady state, solution.

Another test case consisted of a lost foam simulation of a test part. Lost foam simulations are often plagued with convergence difficulties because of frequent global changes in pressures. In the test case, the program selects an initial value of $r_{csql}=4.24e-6$, while previous calculations of this part had been successfully performed using a value of $r_{csql}=0.25e-6$. However, after about 0.25s of simulation time the code has reduced the initial value of r_{csql} to $0.25e-6$ and continues to reduce it throughout most of the computation. At the end of the simulation the value of r_{csql} has reached, or hovers near, its minimum value of $3.85e-8$.

Figure 2 shows that the iteration history and fluid volume error history for this computation. The iterations are clearly at a desirable level (averaging a little below 50), and the volume error is well below 1% after the start up period.

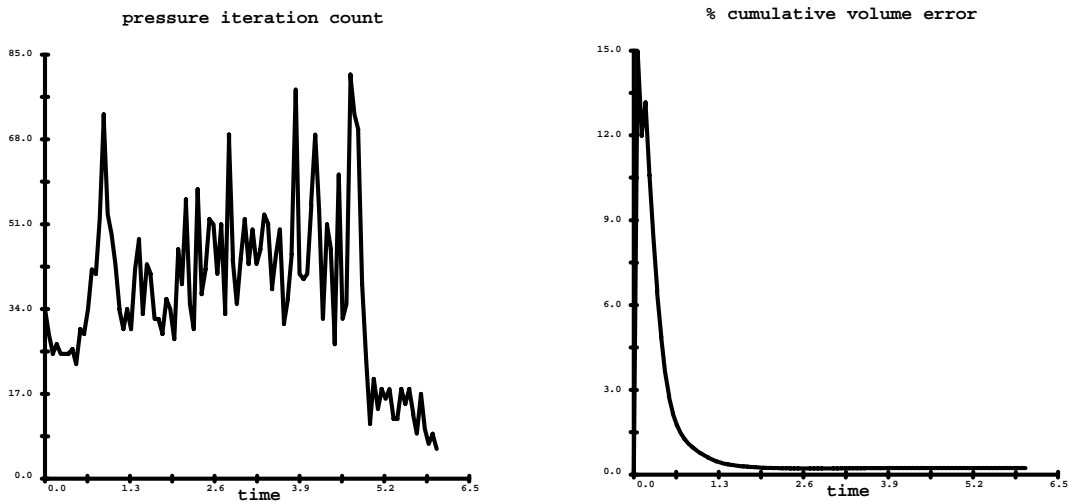


Figure 2. Iteration history (left) and fluid volume error (right) for the lost foam casting using the new automatic limited compressibility feature.

SUMMARY

A new feature to automatically apply and adjust the level of limited compressibility in a fluid has been introduced. All tests have shown the new feature works quite well. This addition significantly improves the convergence of the overall model and eliminates the task of having to experimentally determine a suitable value for limited compressibility.