

## Implicit Advection in *FLOW-3D*<sup>®</sup> : Its Uses and Limitations

5/17/06 CWH

### Overview

A powerful implicit advection technique has been incorporated into *FLOW-3D*<sup>®</sup>, Version 9.2. This paper illustrates uses of this technique to show its advantages, but also indicates certain limitations related to the accuracy of implicit methods.

Using the implicit advection scheme requires the selection of an input parameter *impadv* that activates the scheme according to:

- impadv* = 0, no implicit advection (i.e., explicit, the default);
- 1, implicit advection, with limited advection at free surfaces controlling the time-step size for accuracy;
- 2, implicit advection, with no advection limit on time-step size.

When using *FLOW-3D*<sup>®</sup> to simulate transient problems, especially those involving sharp free surfaces and/or fluid-fluid interfaces the best implicit advection option is *impadv*=1. In this case, the program will limit time-step size by those fluid velocities at a free surface where the velocity is normal to the surface and the fluid fraction at that location has changed by more than 5% in the preceding cycle. Otherwise, the surface velocities will not impose a limit on time-step size.

The second option, *impadv*=2, has no advection velocity limit and should only be used for accelerating an approach to steady-state conditions, but the presence of free surfaces or sharp fluid-fluid interfaces could introduce poor results so this option must be used with caution. If there are no sharp interfaces, then the choice of *impadv* = 1 or 2 makes no difference.

The *FLOW-3D*<sup>®</sup> solver always outputs periodic messages indicating what process is controlling the time-step size. In explicit computations, where fluid advection is the controlling process, the message includes a two-character indicator “cx”, “cy” or “cz” signifying it is advection (i.e., fluid convection) in the x, y or z direction. With the implicit advection option *impadv*=1 advection at a free surface may still limit the time-step size. This is indicated in the message indicator by “sx”, “sy” or “sz”, where “s” is used in place of “c” to emphasize surface velocities.

It is often helpful when using the *impadv*=1 option to scan the “Solver Summary” file listed under the Diagnostics tab. This file contains short prints of all the processes that could control the time-step size. In particular, it includes both the explicit advection limits as well as the implicit advection limits imposed at free surfaces. Comparison of these limits indicates how much larger the time step is when using the implicit option.

It should always be kept in mind that implicit techniques invariably introduce some amount of smoothing or damping of the computed results beyond what is expected in explicit simulations. The following examples illustrate both the advantages and limitations of the implicit advection technique.

### Sample Test for Implicit Advection

A simple two-dimensional test example nicely illustrates the implicit advection option. A rectangular tank 30cm wide and 20cm high is outfitted with two internal baffles, one from the bottom and the other from the top. A small, 1cm, inlet port is defined at the middle of the left side of the tank and a corresponding outlet is located on the right side. The inlet/outlet flow speeds are fixed at 30cm/s and the tank is initially filled with water at rest (i.e., there is no free surface). The purpose of the simulation is to determine steady flow conditions.

An explicit computation using a grid of 60 by 40 elements, Fig.1, required 310s of CPU time to reach near steady conditions at  $t=30s$ . The maximum stable time step size,  $\delta t=0.0033s$ , is controlled by the horizontal flow at edge of the outlet. With the implicit option *impadv*=1 the CPU time was 72s (4.3 times faster than the explicit case) and the time-step size was always at least an order of magnitude larger than the explicit value.

The converged steady-state results are in very close agreement. More iterations were required to get the flow started (including about 17 cases of SOR pressure iteration failure), which reduce the time-step size to the 0.03s range (an order of magnitude larger than the explicit value). Once convergence is achieved and steady conditions are approached, the time-step size increases to 0.1s by the end of the computation.

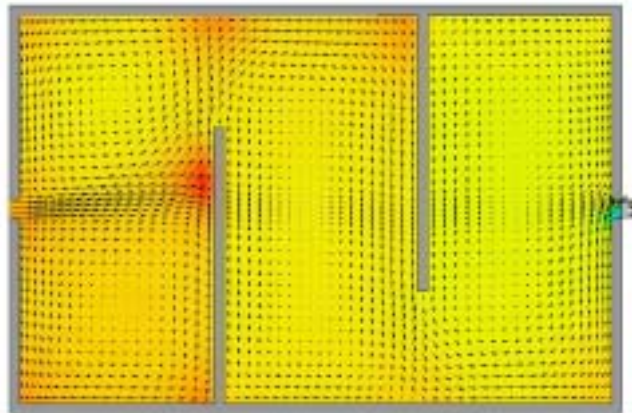


Figure 1. Steady flow in 2D tank with small inlet/outlet.  
The implicit case is 4.3 times faster than the explicit case.

A nice variation on this problem is to add a free surface by not filling the tank completely. Initially the fluid height is set at 17.0cm. For implicit advection, we select the option *impadv*=1 to maintain the accuracy of the computation at the free surface. It is also necessary to use a second implicit option that is for free-surface gravity waves (*impsrf*=1). Without this the time-step size would be smaller than that for advection, making implicit advection unnecessary. Figure 2a shows the computed results, which are

nearly identical to results obtained with an explicit computation. The color shading shows where implicit advection is influencing the inlet jet region, a region at the outlet and in a small region near the upstream base of the baffle hanging from the ceiling. The total region of implicitness is quite small so its effect on accuracy will also be small.

In both the explicit and implicit cases the flow is essentially steady by about 10s. The explicit case required 150s of CPU time to reach  $t=10s$ , while the implicit case only took 92s of CPU time, a speed up of 1.6 times. CPU times to  $t=30s$  were 421s and 168s, a speed up of 2.5 times.

Trying to push this test case faster by eliminating the free surface advective limit ( $impadv=1$ ) does not offer much of an advantage. The time-step size is only about a factor of 2 larger, but more iterations are required keeping the CPU time about the same.

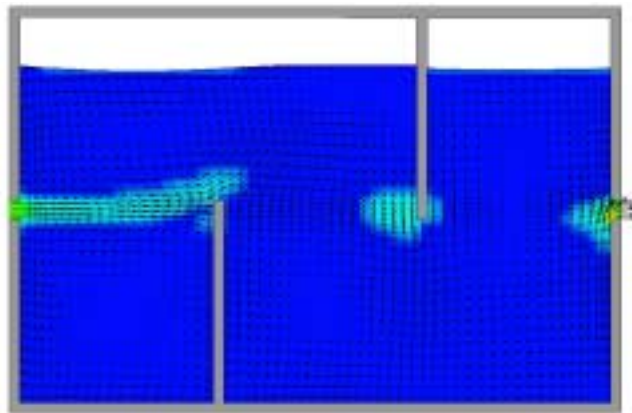


Figure 2a. Steady flow in tank with free surface. Implicitness only used in light colored regions near inlet, outlet and lower end of second baffle.

### Scalar Advection in a Steady Flow

The accuracy of the steady flow in Fig. 2 is not affected by the implicit option since the steady results should be the same as those obtained with an explicit computation. Only transient behavior is likely to be influenced by implicit treatments which involve making changes in the time-levels of terms in the finite-difference equations.

To illustrate this aspect of implicit methods a restart was performed on the previous problem, which has a steady flow established. However, a passive scalar was given a boundary value of 100 at the left, inlet boundary and the computation was carried forward for an additional period of 20s. Fig. 2b shows comparisons of the distribution of the scalar after 2s and after 20s when computed with and without implicit advection.

After 2s it is clear that with implicit advection the scalar has not penetrated into the flow chamber quite as far as it has in the explicit case. This was expected because the nearly 10 times larger time-step size used in the implicit case makes it necessary to limit

advection of the scalar in the high-speed inlet region (i.e., the source region). After 20s the differences are less pronounced, but there still remains some lag in scalar values.

This example emphasizes the slowdown in advective processes with an implicit option because we introduced the scalar source in the worst possible location, i.e., where the flow speed is exceeding the explicit Courant stability condition by the greatest amount. However, as time progresses, the gradient in scalar distribution near the source decreases and the accuracy of the implicit method improves.

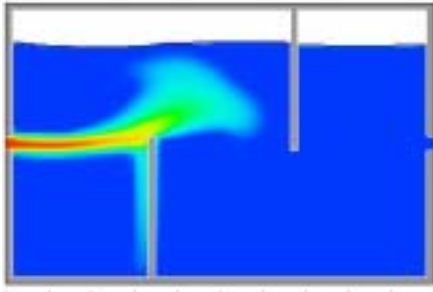


Figure 2b. Implicit advection at 2s.

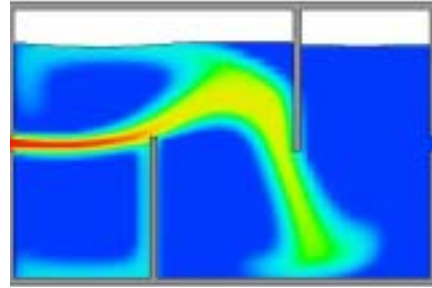


Figure 2b. Explicit advection at 2s.

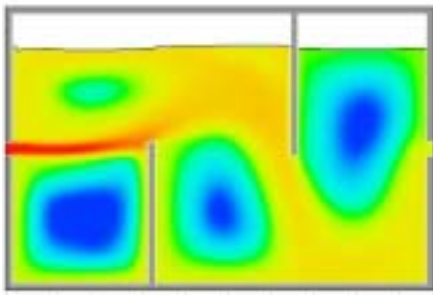


Figure 2b. Implicit advection at 20s.

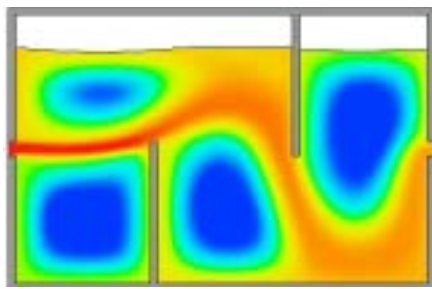


Figure 2b. Explicit advection at 20s.

### Notched Weir Flow

The ***FLOW-3D***<sup>®</sup> Example Problem, “Weir” is a three-dimensional flow of water through a notched weir structure. This problem has been simulated using the implicit advection option *impadv*=1 with results that are nearly identical to the original explicit computation. There is a 16% reduction in CPU time, probably not enough to justify the implicit method. This should not be surprising because the maximum fluid velocities controlling time-step size occur at the fluid free surface. Everywhere away from the surface the flow is slower and no implicit advection is necessary to insure stability.

This is a good example of a situation in which implicit advection has no effect on a simulation. As long as accuracy is demanded, there is an upper limit on the time-step size that can be used. At free surfaces, for example, the time step must be limited to the usual advective Courant condition. Attempting to use higher values usually results in distortions or other inaccuracies in the flow. This is illustrated in the next example.

### Collapse of a Water Column

A standard free surface problem is the collapse of a column of water. In this two-dimensional example a water column 1 cm square is located in the lower left corner of a tank that is 2.5cm wide and 1.2cm high. Gravity is down at  $980.0\text{cm/s}^2$ , which causes the water to slump and flow out along the bottom of the tank.

Figure 3a shows the resulting flow configuration at  $t=0.054\text{s}$  when run explicitly. As in the preceding case, the largest velocities are at the surface and the implicit adjustment is rarely invoked when the implicit option *impadv*=1 is used.

When *impadv* =2 is input, so that larger time-step sizes are used (i.e., there are no advection limits at the free surface), the results at the same problem time are shown in Fig. 3b. Color shading indicates the fluid velocity magnitude. The front is the fastest moving region, but is seen to be moving slower in the implicit case. Implicitness has added some dissipation to the flow slowing it down. The implicit CPU time is only about 17% of that required for the explicit run, but the reduction in accuracy may not be worth this reduction in CPU time.

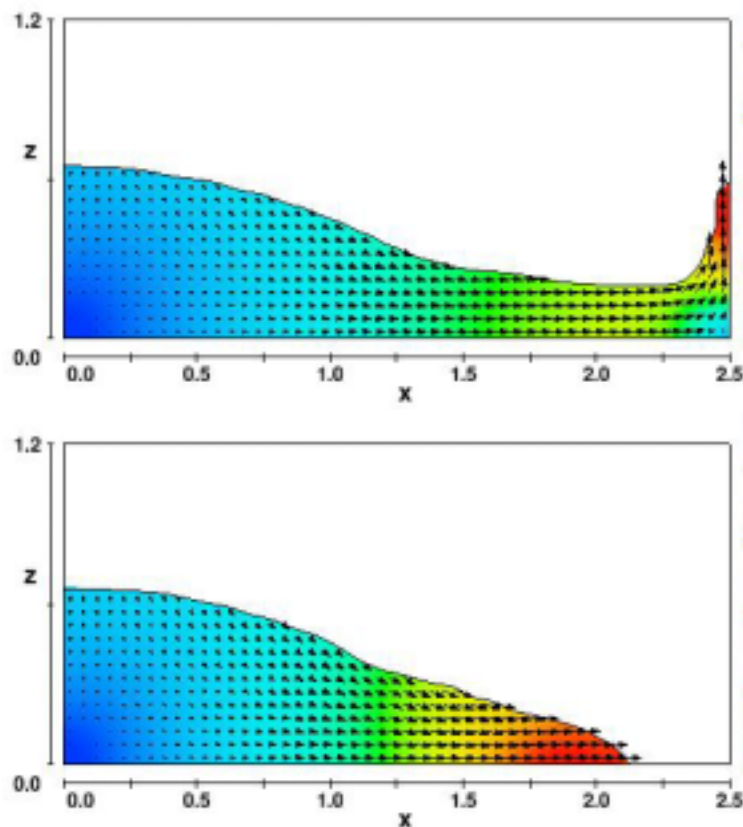


Figure 3. Top (a) accurate depiction of collapsing column of water, bottom (b) same problem with full implicitness showing slowdown of front.

### Additional Examples – Box Cast, MoldThrm and Stir

In the list of **FLOW-3D**<sup>®</sup> Example Problems there are several that might profit from implicit advection. In Example Problem “Box Cast” is an early high-pressure aluminum die cast involving a thin-walled box shape. Simulating the filling of the die, using the *impadv*=1 option for an accurate free surface, gives quite acceptable results and reduces the CPU time to 49% of that required by an explicit simulation. Although expectations for high-pressure die casting were for a much larger savings in CPU time, this test case is atypical because the thickness of the gate where metal enters the die is the same as the thickness of the box wall. In most high-pressure die casts the gate is smaller and experiences the highest flow velocities, a situation where implicit advection is expected to offer a significant advantage. However, even the speedup of a factor of two obtained in this example can be considered worthwhile.

MoldThrm is another casting Example Problem. Implicit advection does not offer much improvement in CPU time (about 18%) since during much of the time the free surface velocities are, or are close to, the maximum fluid velocities.

The Example Problem “Stir”, which stands for stir tank, does not involve a free surface. This is basically a three-dimensional steady flow problem involving a rotating impeller and a set of four baffles attached to the wall of a cylindrical tank. An explicit simulation to a problem time of 10.0s has a time-step size limited to about 0.007s. With the implicit option *impadv*=1, the average time-step size starts at least a factor of 10 greater than the explicit value. After about 3s of simulation time, when steady conditions are being approached, the step size increases linearly to about 56 times the explicit value at the end of the computation. The total CPU time to reach t=10.0s is about 4 times faster than the explicit case.

Because only steady conditions are wanted there are additional options that could have been selected to accelerate the computations. For example, using a larger convergence criteria to reduce the iteration number, say *epsadj*=10.0, reduces the number of iterations and, correspondingly, the CPU time is 6 times shorter than the explicit case. Even so, a factor of 4 speed increase is significant and was achieved without having to experiment with other parameters. Another point is that even though the time-step size is at least 10 times larger than the allowable explicit value the CPU time reduction is less than this because more iterations are required with the larger time step.

### Flow Over a Cube – A Steady-State Example

As an example of steady flow we have used a simple problem of flow over a cube. In non-dimensional units the cube has unit edge length and is set on the bottom of a rectangular flow region of length  $x=7.0$ , width  $y=2.0$  and height  $z=4.0$ . Symmetry is assumed in the  $y$  direction and a specified flow of  $u=30.0$  is defined at  $x=0$ . At  $x=7.0$  a continuative boundary is used, while all other boundaries are planes of symmetry. To approximate turbulent conditions an eddy viscosity of 0.03 was defined, i.e.,  $0.03(\text{edge length})(\text{flow speed})$ .

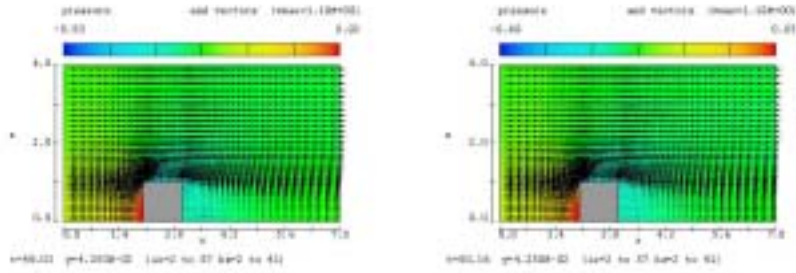


Figure 4a. Flow in X-Z symmetry plane, explicit run on left.

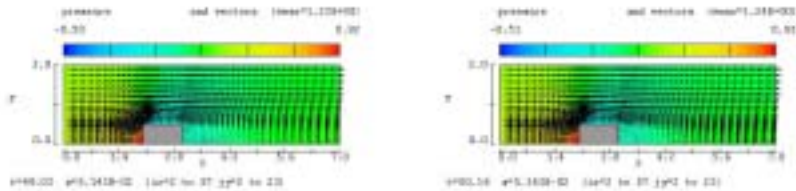


Figure 4b. Flow in X-Y ground plane, explicit run on left.

Because of the eddy viscosity the time-step size is controlled by viscosity, so that implicit advection offers no advantage. For this reason we used an implicit viscous option, *impvis*=1 and *itvsmx*=3 to eliminate the viscous limit on the time step.

Two simulations were run; one was explicit with *epsadj*=10.0 and the other was identical except for implicit advection. No limited compressibility was used to accelerate pressure convergence. In both cases a converged, steady solution was observed to develop at about 20 time units into the problem. The CPU times to reach  $t=20.0$  were 1600 and 413, giving an implicit speedup of a factor of 6.4.

The explicit simulation was run out to a time of  $t=49.0$ , which took 3500s of CPU time. The corresponding CPU time for the implicit case was 338s, or a speedup of a factor of 10.4. In both cases the pressure iterations did not converge for the first 30 cycles or so, but this is of no consequence since only steady results were of interest.

Figures 4a-b show comparisons of the results at about  $t=49$ . Details are hard to see, but the pressure ranges and maximum flow speeds are in very good agreement.

### High-Pressure Die Casting – NADCA Part

With great expectations a simulation of an actual high-pressure die cast part was simulated. A non-proprietary alternator housing, referred to as the NADCA part, was selected for this test. The expectation was that in high-pressure die casting there is generally a thin gate region where the largest flow velocities are recorded and that this flow region is relatively steady during the majority of the die filling process. Such a situation should be ideal for the implicit advection technique to show a significant speedup without loss of accuracy.

In this example the gate region does not completely fill, see Fig. 5b, until the very end of the simulation. Thus, the time-step size for accurate free surfaces with implicit advection

must take into consideration the free surface velocities at the gate, which means that a large speedup might not be possible. In fact, because we have some restrictions on the free surface velocities that must be treated explicitly, namely that they are normal to a surface and that the local fluid fraction must be changing by more than 5% in a cycle, the actual limitation is not so bad. The free surface located at the gate arises from a flow separation that is relatively stationary so the time-step size does not have to be limited much to maintain accuracy. In fact, the implicit computation uses a time-step size that is roughly 5 to 10 times larger than the explicit time-step size, and the overall CPU time is about 58% of that needed for the explicit computation. Not a huge increase, but a factor of nearly 2 in speed is certainly worthwhile.

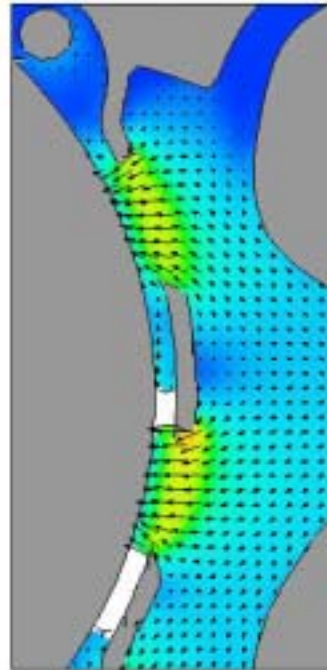


Figure 5a. NADCA part (left) at end of filling. Color indicates temperature.

Figure 5b. Close up of gate region controlling the time-step size.