FSI-97-TN46

SENSITIVITY OF COMPUTATIONAL RESULTS TO DIFFERENT COMPUTERS

Flow Science, Inc.
June, 1997

Overview

It has been observed that different computational results may be generated on different computer systems. The obvious question is whether or not this indicates an error in the software, an error in the compiler that translates the software for operation on a computer, or can there be some other explanation?

In the case of transient free-surface hydrodynamics, a flow occupies a changing set of numerical control volumes (e.g., in FLOW-3D the grid is fixed and fluid interfaces are tracked through this grid). Some criterion must be used to decide (numerically) whether or not an element contains fluid. In FLOW-3D fluid regions are tracked in terms of the fractional volume of fluid in each element. An element with less than one millionth of its volume occupied by fluid is considered empty.

Numerical differences generated by the inclusion or omission of a control volume are on the order of the element size, because the application of free-surface boundary conditions would then differ in location by one element. In other words, a difference of one part in a million in one quantity (the fluid fraction cutoff) can grow into a larger difference characterized by the size of the discrete elements. This is the nature of discrete simulation methods.

Why then do most computations appear nearly identical when run on different computers? The answer lies in the type of problem being simulated. Most flows exhibit smooth and stable behavior. In these cases small perturbations are continually smoothed out and result in no long term consequences.

Unfortunately, there is a class of problems involving fluid dynamic instabilities, which do no exhibit such nice behavior. For this class, by definition, perturbations grow. If there are no limiting constraints, perturbations may grow to such an extent they completely dominate the flow behavior. A glass of water turned upside down offers a simple example. If the surface of the water is sufficiently smooth and flat there will be a little delay before the water starts to leave the glass, but once started it quickly runs out. The smaller the perturbations on the surface, the longer the water remains in the glass.

Numerically simulating an unstable flow problem is a particularly difficult challenge. Generally, when researchers study unstable flows with computational methods they impose initial conditions that set up the mode (or modes) of behavior they are most interested in watching evolve.

If no perturbation is supplied as an initial condition, then one must be introduced by the program to get the flow started. This perturbation might come from an asymmetric pressure iteration, lack of pressure convergence, an asymmetric grid, or other sources.

Another source is machine round off error. As mentioned earlier, differences can arise from the one-part-in-a-million test of whether or not a control volume contains fluid. Although modern computers use long word lengths, round off errors can still reach the one-in-a-million level when you consider that millions of floating point arithmetic operations are being performed in any fluid dynamic simulation.

In particular, the pressures and velocities in an incompressible flow problem are obtained from the solution of a Poisson equation, which is notorious for introducing numerical errors. This type of equation requires the computation of successive differences between pressure values, an operation that emphasizes small numerical differences.

If round off can be a problem, then the fact that different computer platforms may produce somewhat different answers for unstable flow problems should not be surprising. Each computer maker has its own compiler for the software it runs. Compilers invariably differ in the manner and order in which they collect terms and perform arithmetic operations. They can also differ in their treatment of round off for floating point numbers. From this point of view, it would be more surprising if all computers produced identical results.

In the next two sections we illustrate the type of computational differences one can expect when solving inherently unstable flow problems. Two examples are discussed. The first example, a toilet flush, illustrates what can happen when the flow characteristics are strongly dependent on unstable behavior. The second example, flow over a sharp-crested weir, contains a combination of stable and unstable behavior.

Illustrative Example One - Toilet

When a toilet is flushed, water is released into the bowl. This raises the water level forcing flow to flow over a bend at the top of the trap. For a good flush, with enough strength to empty the bowl, water must close off the trap from the outlet to establish a siphon action. Usually, the closure involves the trapping of both water and air in the trap, but bubble volumes remain relatively constant because the change in bubble pressure with respect to volume is large.

Flow in the trap is generally very complex, as it contains a highly turbulent mixture of air and water. It is this region that is difficult to numerically simulate. The complete details of flow in the trap are beyond the fastest computers, and even if they could be computed there is no real interest in such detail. What is wanted is a good estimate of when a siphon action is established. Unfortunately, this is a statistical property that depends on unstable flow processes (e.g., where flow separates from the trap wall and how it responds to air shear forces).

Experiments confirm this statistical property for it is not uncommon to see ±10% variations in flow properties between successive flushes.

We illustrate the computational complexity of this problem using FLOW-3D running on a variety of different computers. Our measure of comparison is the time history of fluid volume in the computation, since it represents a kind of global measure of flushing. For present purposes, the details of the toilet-trap shape and other operating parameters are unimportant. What we are most interested in here is the reproducibility of the computational results on different computer platforms.

Two sets of computations were done for some of the platforms. For the first set, the same identical input data was used for all the computers. In the second set, approximately twice the number of grid cells were used in order to test the sensitivity of results to resolution.

Figure 1 compares seven cases. These cases include the following computers and whether they used low or high resolution:

PC	High & Low
DEC Alpha	High & Low
SUN	High & Low
HP	Low.

During the first 3 seconds all the computations were nearly the same. No siphon action develops during this early time.

For longer times, all the computations, except the SUN running Lower resolution, produced similar results. The SUN case with higher resolution shows results that fall within the general grouping. Most importantly, no two results are identical.

One can also see in Fig. 1 that the higher resolution PC case begins to lose fluid a little sooner than the majority of cases, but this difference is within a ±10% range that might be expected in actual tests.

When details of the different computations are compared it is found that the differences, as expected, are initiated at the time at which fluid spans the entire cross section of the trap to begin the siphon action. This is realistic behavior, because closure in a real trap depends on the details of the turbulent flow in the trap.

Illustrative Example Two - Weir

This example is somewhat simpler than a toilet. The most important feature here, however, is that the upstream flow and flow rate over the weir are stable flow processes. Downstream of the weir, where the overflow strikes the lower water surface, the flow is quite turbulent, Fig. 2a.

Figures 2b-2d show comparisons of different computations run for this problem. In this case, the computations were performed on the five machines (PC, DEC Alpha, SUN, SGI and IBM), but only one grid resolution was considered. All computations used the same input data.

In Fig. 2b we show the computed flow rate over the weir. All the results are identical. This is comforting since it shows that stable flows can be computed confidently on different computer systems.

Figures 2c-2d show comparisons of the computed volume of fluid history and average kinetic energy history. In these plots all the systems produced the same results, except for the DEC Alpha. Comparing details of the computed flows shows that the DEC Alpha exhibits some small deviations in the turbulent, downstream region.

Flow in the weir problem is not as chaotic as in the toilet, and the downstream turbulence cannot affect the upstream flow because flow over the weir is super critical. Consequently, small variations in the downstream flow are usually no cause for concern. Nevertheless, this example demonstrates once again that chaotic flow regions governed by fluid dynamic instabilities may not be reproducible on different computers.

Summary Comments

When comparing computed results between different computer platforms, you should expect some differences in round-off errors that may lead to differences in the location of free surfaces within a grid. If the flow is chaotic, not smooth, these differences will usually grow rather than diminish. When this happens different results can be obtained on different computer systems. These differences can be loosely interpreted as different realizations of a turbulent flow.

If the computational results one is interested in are affected in this way, then one has to be careful not to accept the computations as absolute. In such cases it is important to perform a sensitivity analysis, for example, to refine the grid a little and see how much the results change. Other possible variation tests could include a tightened convergence level (i.e., smaller EPSADJ) or switching on the higher order momentum advection approximation (i.e., IORDER=3).

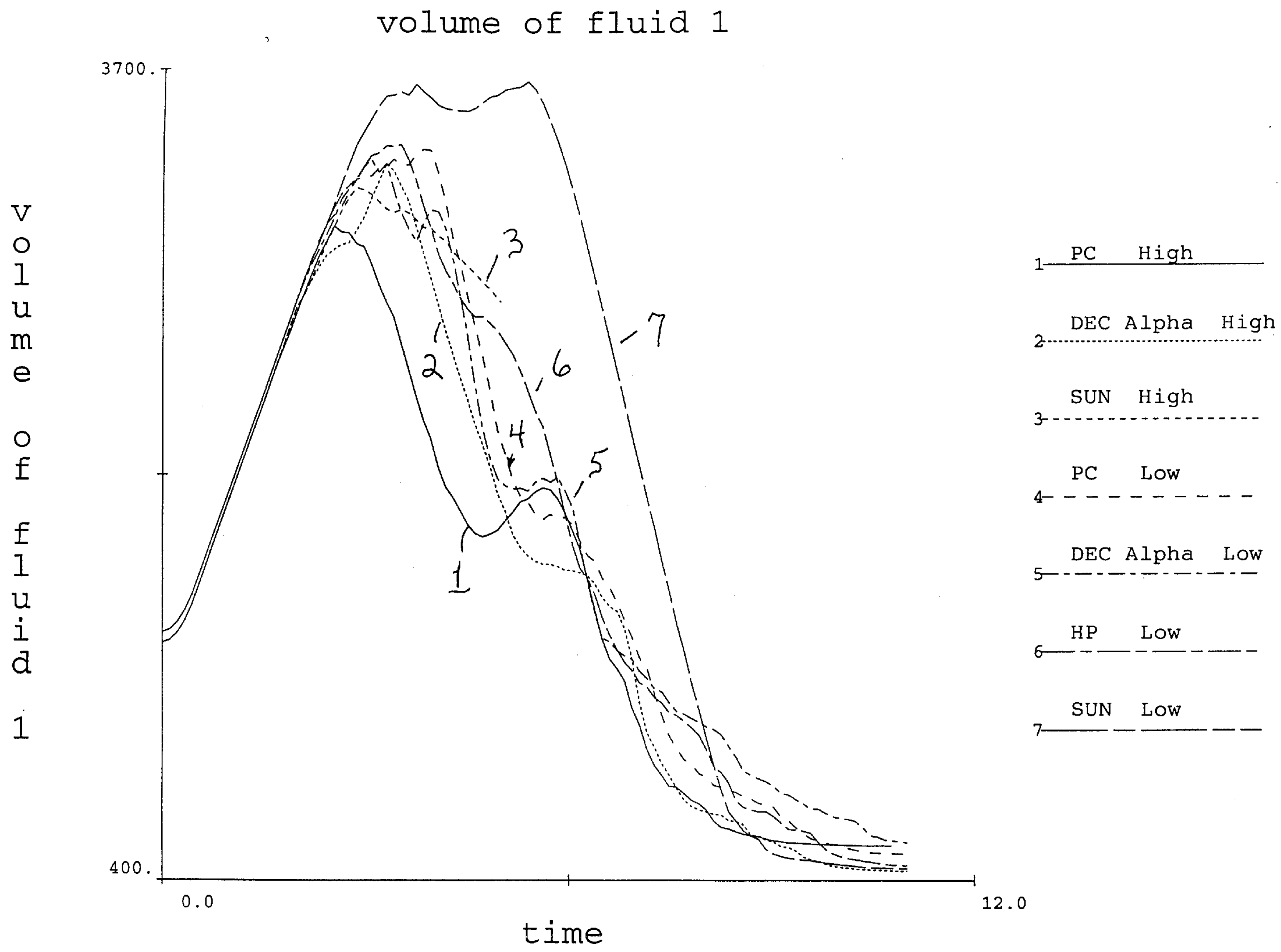


Figure 1. Comparison of results for toilet flush test.

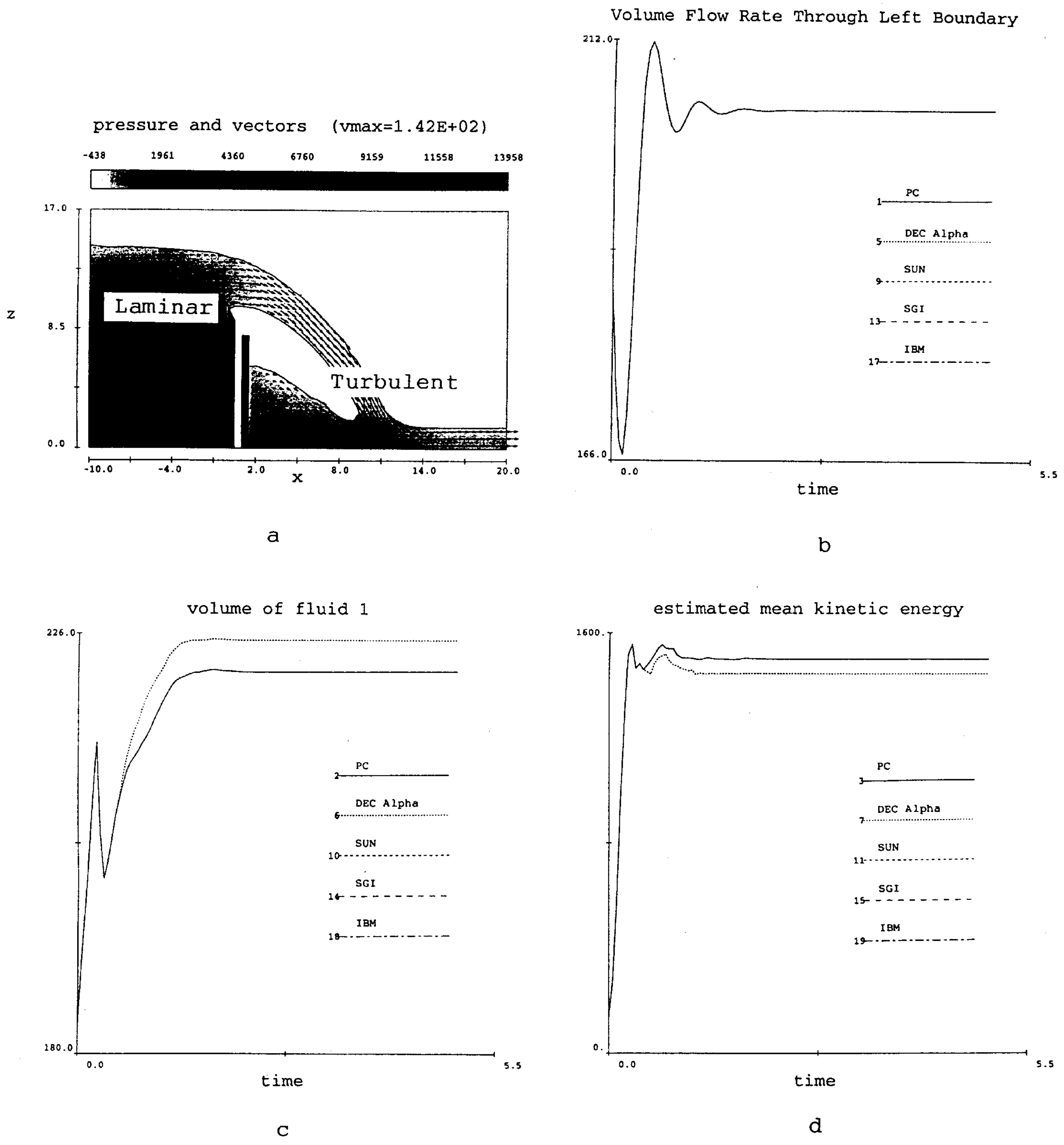
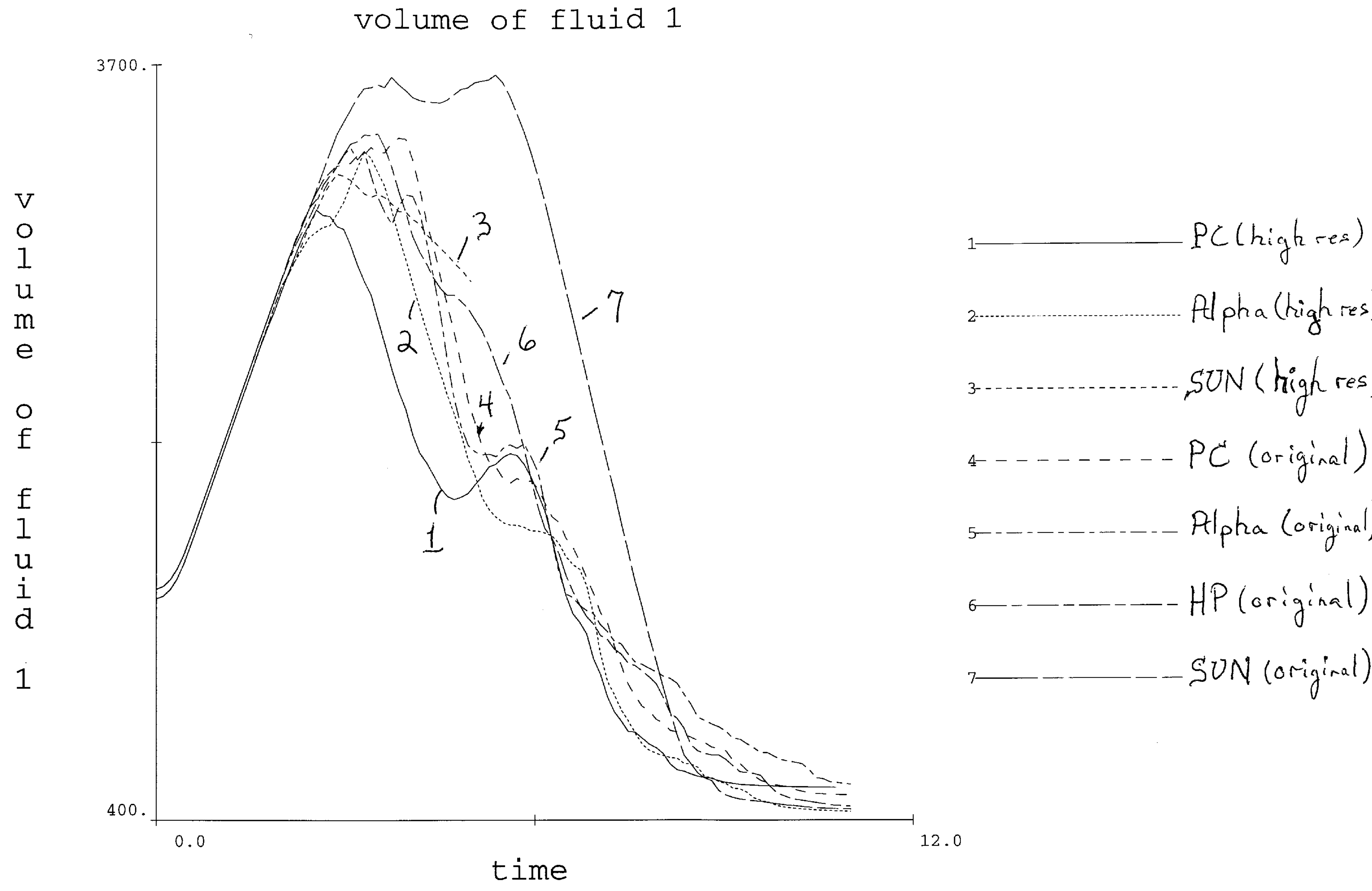
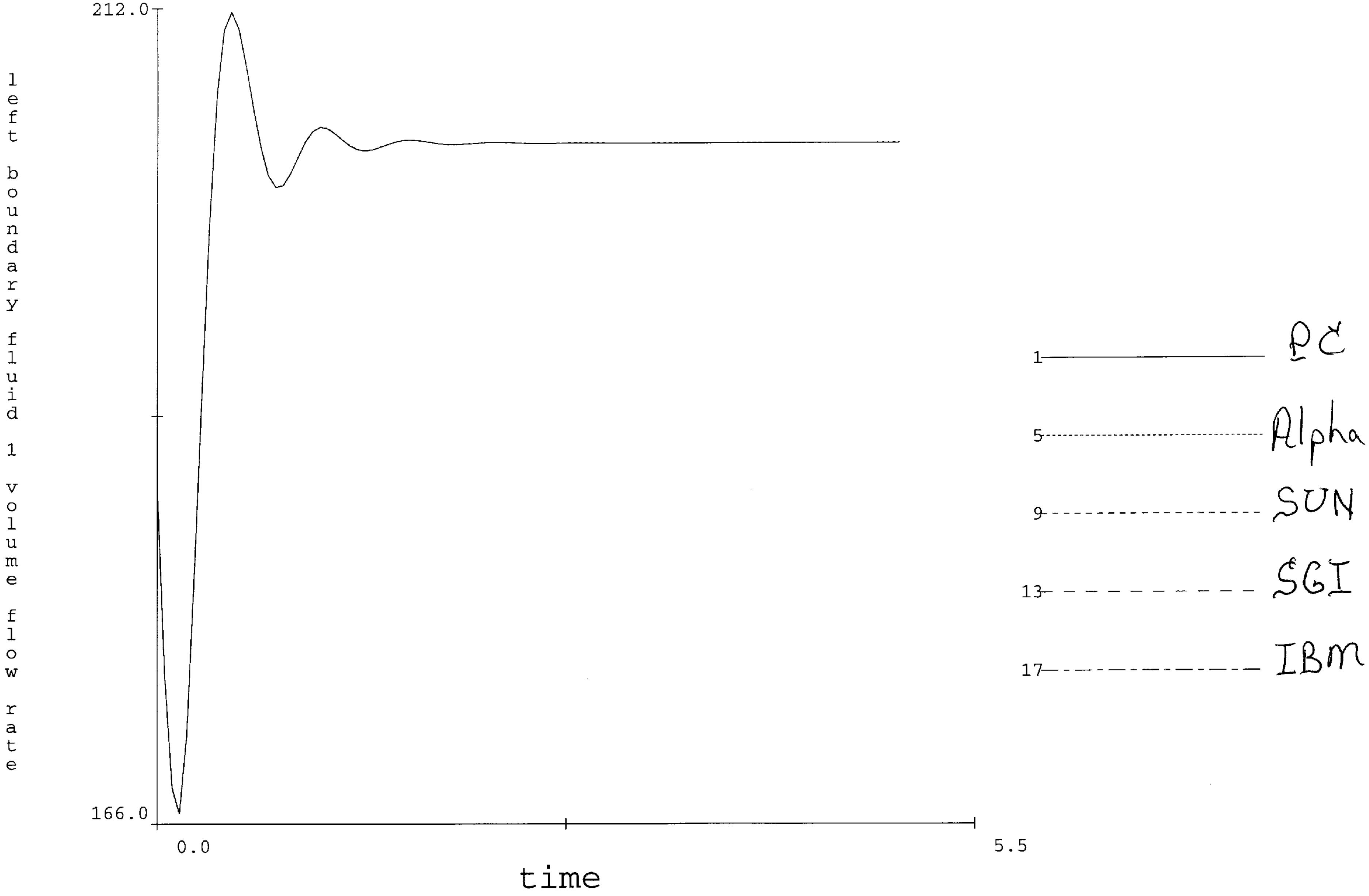


Figure 2. Flow over a sharp crested weir.

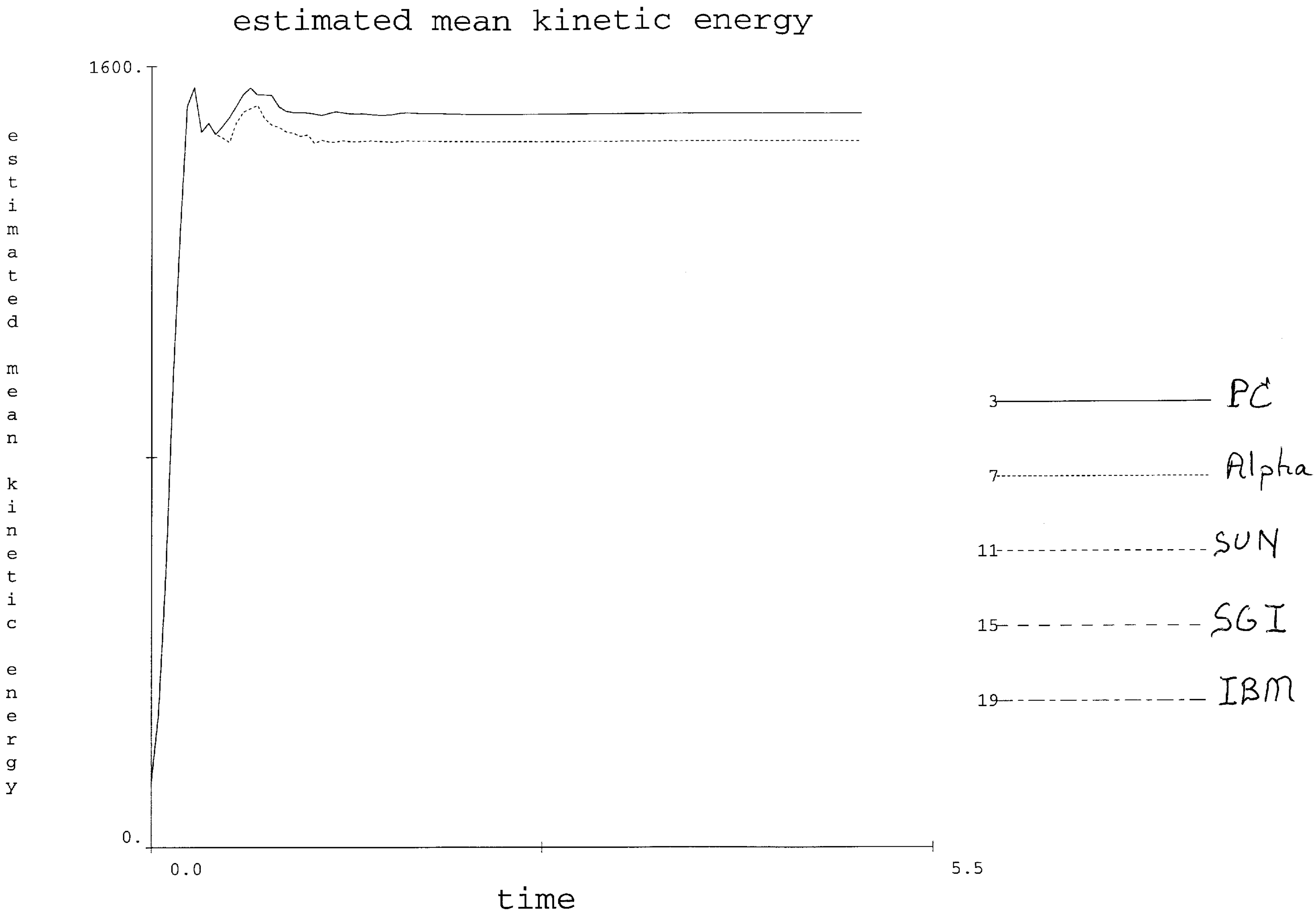


FLOW-3D®
16:28:09 6- 9-1997ponm hydr3d: version 7.0.14 win32 1996
Simulation of Toilet Flush

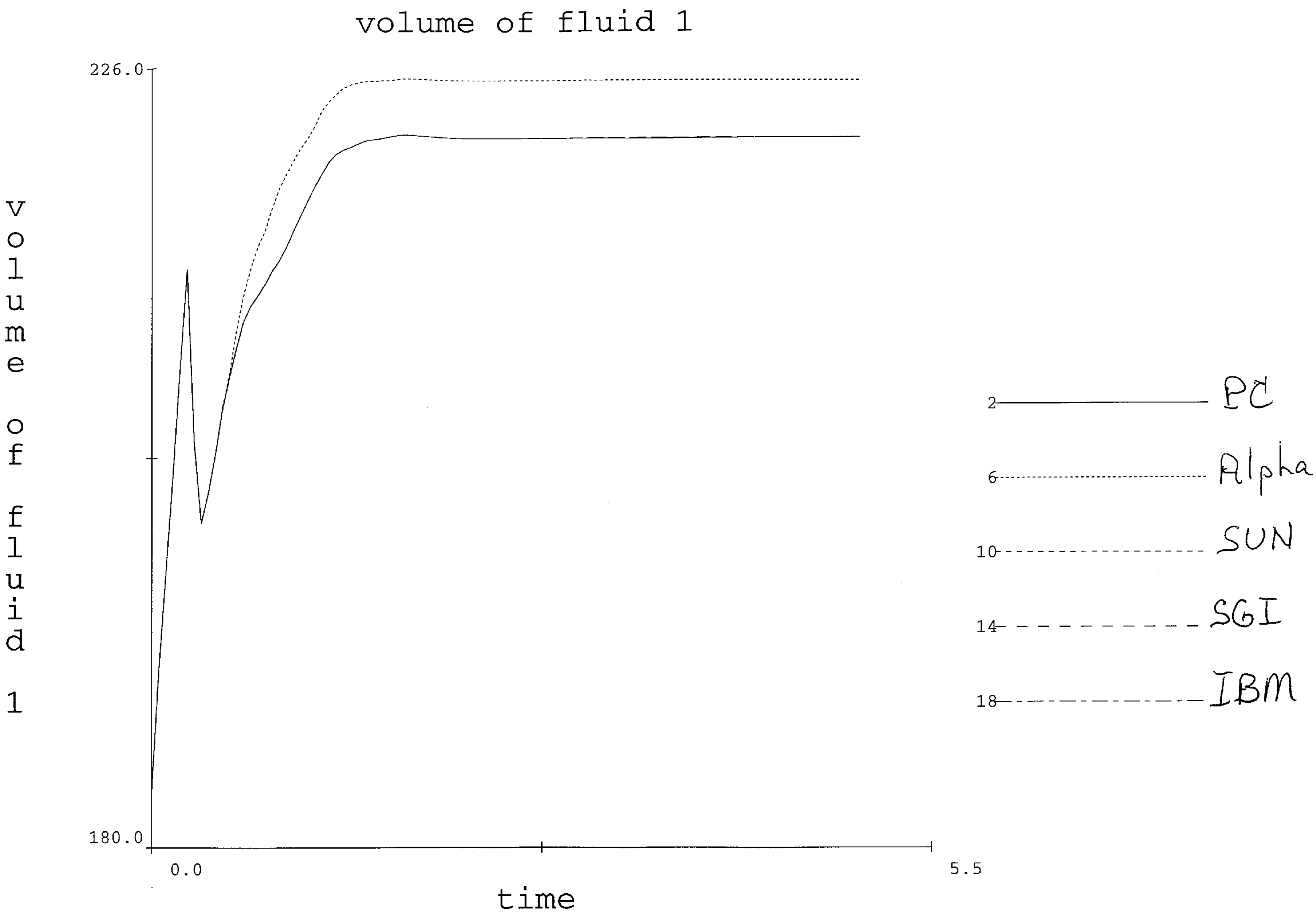
Volume Flow Rate Through Left Boundary



FLOW-3D®
13:28:16   6-11-1997mtfh   hydr3d: version 7.0.14   win32 1996
flow over sharp-crested weir -- g=980, RNG, #3



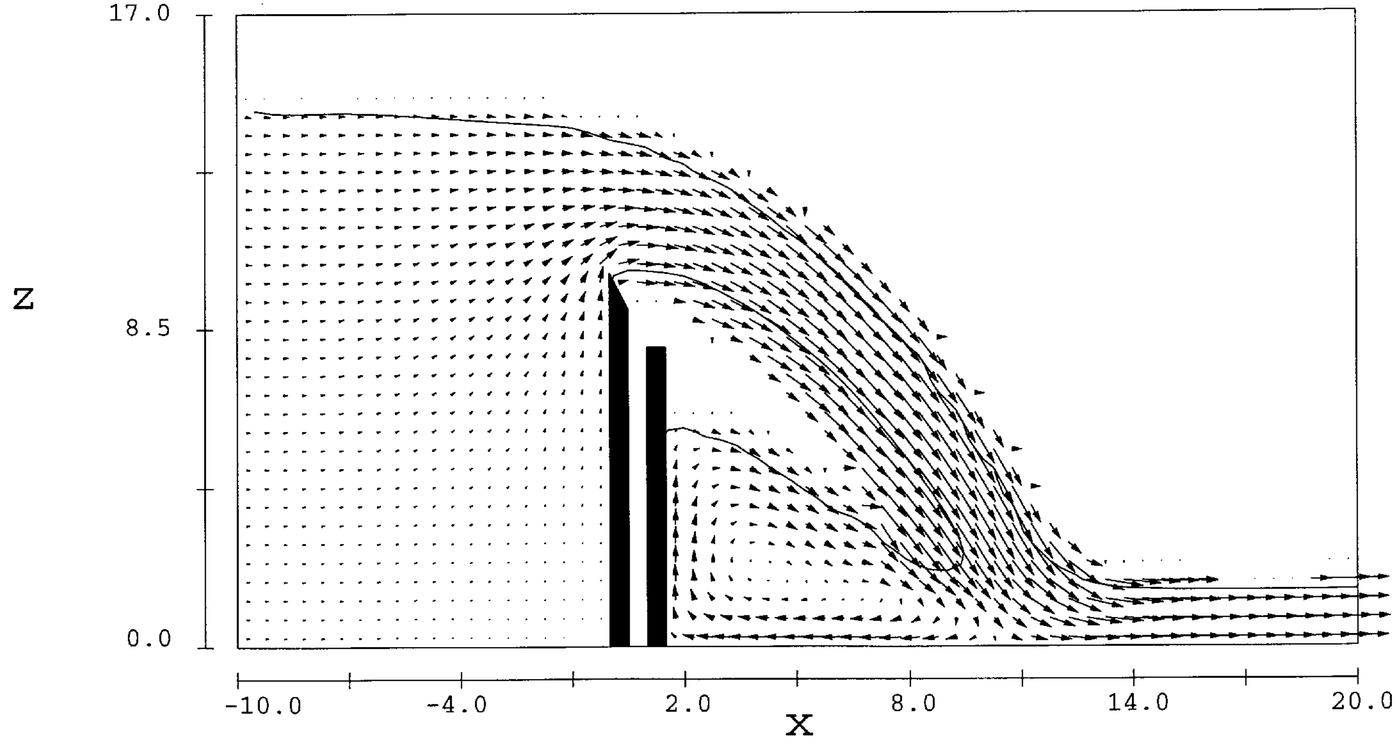
FLOW-3D®
13:28:16   6-11-1997mtfh   hydr3d: version 7.0.14   win32 1996
flow over sharp-crested weir -- g=980, RNG, #3



FLOW-3D®
13:28:16   6-11-1997mtfh   hydr3d: version 7.0.14   win32 1996
flow over sharp-crested weir -- g=980, RNG, #3

velocity vectors

(→ = 1.42E+02)

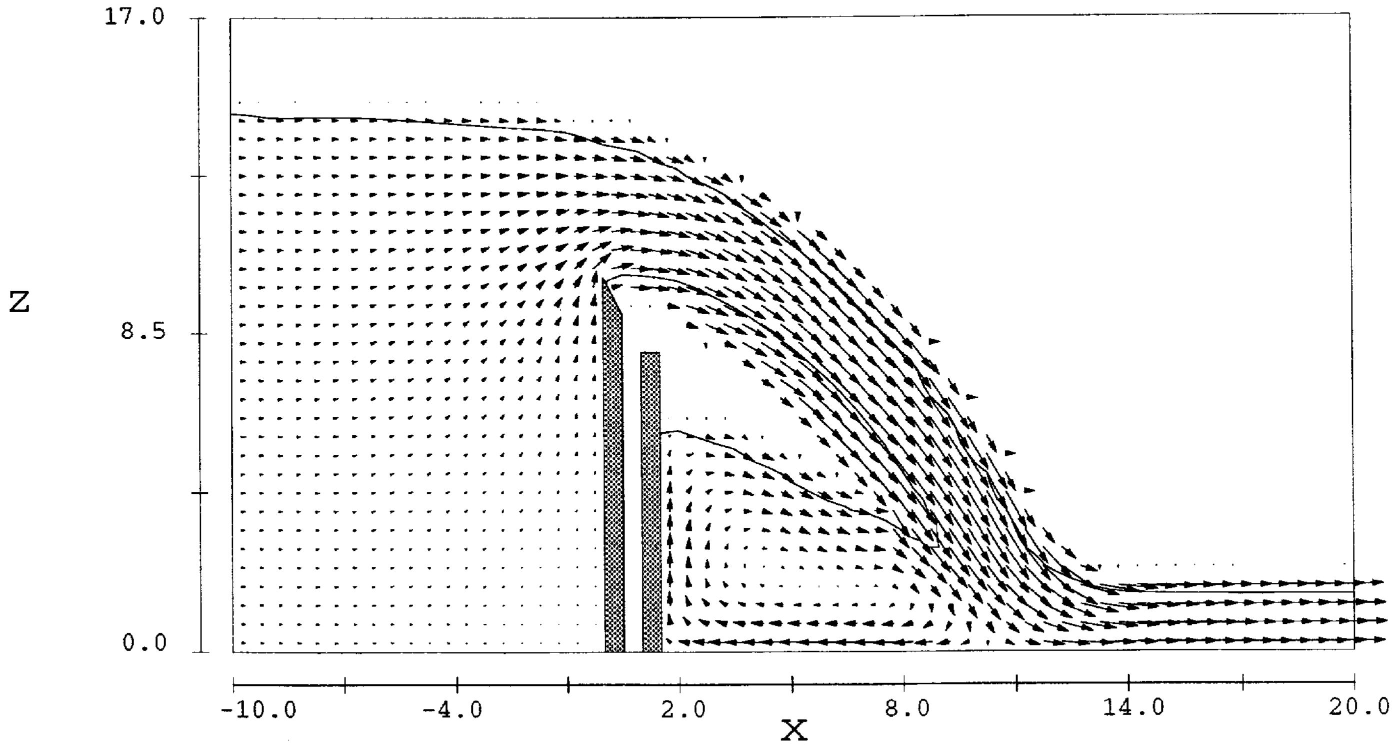


FLOW-3D® t=5.001 y=5.000E-01 (ix=3 to 62 kz=3 to 36)
13:28:16 6-11-1997mtfh hydr3d: version 7.0.14 win32 1996
flow over sharp-crested weir -- g=980, RNG, #3

DEC α

velocity vectors

(⟶ = 1.38E+02)



FLOW-3D® t=5.001 y=5.000E-01 (ix=3 to 62 kz=3 to 36)
17:16:0400 6/11/97 eifd hydr3d: version 7.0.14 alpha 1996
flow over sharp-crested weir -- g=980, RNG, #3