Partial 3

Andrés Felipe Gómez

May 2019

1 General Instructions

Some minor changes must be done to the principal programs in order to run them to generate as results the simulations in general form; a makefile was designed for the specific cases.

1. Total time of integration and integration step can be changed in the program main.c just before the routine evolution.
2. The directory to save the output files from main is indicated and can be changed in the programs ics.c and rutinas-sph.c (by defaults it is data1/punto1), in the routines genera-condiciones-iniciales-mayaN and evolution respectively. Where N represents a number that indicates a different routine for another set of initial conditions.
3. Genera-condiciones-iniciales-mayaN can be changed for other routine to generate the initial conditions in main.
4. The directory to read the output files to generate the simulations must be changed in plot-fluid.plt or plot-velocity.plt; by default it is data1/punto1/.

For point1: How to run the simulation to generate a video of the result?

I set in the following lines a not strict order that could be used:

1. Change total integration time *Totaltime* and step of integration *dt* in main file.
2. Chose a routine to generate the initial conditions; genera-condiciones-iniciales-maya2 is available for the present purpose.
3. Once selected this routine, densidad1 is chosen for the fluid for y's between 0 and 0.5 and densidad2 for the fluid between 0.5 and 1; both with the initial velocity equal to 1; or the one specified in ics.c in the routine. genera-condiciones-iniciales-maya2.
4. Indicate the directory to save output files from main and read them as specified in general instructions.
5. Finally run ./sph2 Number-of-particles to generate the output file from main.

In this case Number-of-particles must satisfy that sqrt(Number-of-particles/2)/2 is integer. This to give symmetry to the problem and simplify calculations. 6. Run then plot-fluid-principal.gp plot-velocity-principal.gp ARG1 ARG2; where ARG1 is the directory where the output frames are saved as well as the video. The frames of the velocity maps and its video are saved in ARG1/velocity and the frames of the density map and its video are saved in ARG1. ARG2 is the number of frames to be used for the creation of the video.
7. Enjoy the result.

For point2: How to run the simulation to generate a video of the result?

Only change step 2 by using the routine genera-condiciones-iniciales-maya3 and 4 if the output is desired in another place. Also Number-of-particles must satisfy sqrt(Number-of-particles/2) is an integer.

Designed example

To test the program the following example was set. Do make make run1-ver1 as an example, the output videos for the first example explained in the following lines is generated.

2 Analysis for point 1 physical discussion

To give a complete analysis it must be taken into account the fact that the simulations have considerable errors due to the limitations that has the methodology used. To study how this one can affect the result several computational experiments were carried out.

1. Empirical analysis to choose a step time for a given number of particles.

The fluids were ran for 1024 and 4096 particles with 64 neighbourhoods, and they were found to behave in a not physical way. The files in the directory results alone1024.mp4 as well as alone4096 illustrate the situation. This is of course a not real physical situation. In fact there is a considerable number of particles that move to the left, fact that has not sense at all.

Now, from the fact that the particles behave worst with 4096, that is, they go away in less time and get more negative densities, we can suspect of a problem with the integration time as depending of the number of particles. After such consideration it was chosen to run for a number of particles that under the integration time 2e-3 and dt = 1e-3 do not get negative densities. A suitable number was found to be 100 particles.

For such number, the simulation were ran over the parameters $\rho_1 = 1.2$ and $\rho_2 = 997$, $\beta = 0.01$, $\alpha = 1$, $\nu = 0.01h^2$, $cs_1 = 1493$, and $cs_2 = 343$ in an attempt to simulate water and air. The first results are found in the directory result and are named animationrhoP1N100t2Em4dt1Em6.mp4 and animationvelP1N100t2Em4dt1Em6.mp4.

Several features can be observed to be reproduced as well as some that clearly not. First the density is always positive, the movement of all the particles is to the right.

In the following I try to explain some of the features observed, possible errors and some considerations.

1. The fact that viscosity is greater in water it is clear from the fact the particles of air try to follow it as it was in some degree a rigid wall; so the general behaviour expected for the air is in some degree reproduced (nevertheless point 5 below also have an influence in this).
2. The symmetric "peak" observed around 0.3 could be cause by the fact that the density is interpolated and such interpolations make less dense particles around the boundary with water and with the vacuum; and as response to the viscosity such regions decelerate more rapidly, thanks to the fact that such are less massive particles.
3. To try to illustrate the last point a simulation with 144 particles was ran and the results animationrhoP1N144t2Em3dt1Em7.mp4 animationvelP1N144t2Em3dt1Em7.mp4

(in directory results) illustrates best the point and the symmetry around 0.3.
4. The fact that some particles go away can be related to the fact that some bulk motion is generated and as the acceleration depends upon how big is the gradient in density, the fact that is not well mapped make the some particles to accelerate in an unreal way, thanks to a great density gradient that should not exist. In fact, note that taking away the more to the right particle in animationrhoP1N144t2Em3dt1Em7.mp4 the observations in 3 are more clear.
5. There is an inclination in the code; probably from the kernel to make the particles to move more to the right, fact that introduce a systematic bias in the simulation.
6. The not consideration of boundaries introduce considerable errors when not a great number of particles is used as was illustrated before.
7. A not well mapped region create nonphysical situations.

3 Analysis for point 2 and physical discussion

First of all, something must be done in order to try to eliminate the bias of the velocity, so supposing this one is at most of 0.5 from the observations; we would like the bound the error in one step, such that if we do 1000 steps, the particles get at most affected because of it by 10 percent approximately. Therefore, we choose v = 10000 and v =-10000 for water and air with the same parameter chosen in the example above.

After such considerations several number of particles were tried for integration steps of 1e-6 and total time of 1e-3. So, a good approximation was obtained by using a number of 200 particles, the results are given by animationrhoP2N200t1Em4t1Em6.mp4 and animationvelP2N200t1Em4t1Em6.mp4 in directory results. It is note that the interaction of the particles was not as expected in the sense that could be expected that greater number the particle finish having velocities near 0 after the collision.

So, to reproduce this behaviour a simulation over the total time 5e-4 was ran, but with a step time of 1e-7 and 512 particles. The result are illustrated by animationrhoP2N512t5Em4dt1Em7.mp4 and animationvelP2N512t5Em4dt1Em7.mp4. Although there is a great variation in density it is observed that the particles after the collision reduced greatly their velocity.

In general the observations 2, 4-7 did before are still valid in some degree and response in some degree by the result of the simulation.

4 Conclusions

To make a good sph code it is necessary to have into account the regime in which the fluid is expected to move and to secure first expected and basic behaviors.

It is not that it is not a trivial matter to do a good sph code, but that systematically improvements can be made based in a good physical background.

The integration step was found to be very sensible to the number of particles, one way to improve the code could be to simply improve the integration step, nevertheless, another thing that could be done is the implementation of a best kernel, that could permit to detect spurious interactions; although this could introduce a correlation this one could be well studied.

A good improvement to made is the use of a gaussian kernel, because as is said in Monaghan-1992.pdf it is better for the analysis of physical situations.