6. Conclusion

We have vectorised the SPH algorithm, which now is quite time-performant with respect to other concurrent hydrodynamical codes. The algorithm has the great advantages of being Lagrangian and of simple conception. Also the code introduces only little dissipation, in contrast to grid-dependent algorithms. In our vectorised code, the CPU time grows linearly with the number of particles. This is due to the constant number of neighbours necessary to ensure good accuracy and to the algorithm that suppresses all unnecessary calculations. This enables us to handle many more particles and then to improve significantly the dynamics, which was a weak point of the SPH algorithm, although somewhat cured by the variable resolution. The larger number of particles also allows a correct treatment of the low-density regions, such as the boundaries, which was a second difficulty of the method.

In our code, it is to be noted that the calculations of the macroscopic variables and the gradients are much more efficiently vectorised (factor 20), while for the search of the neighbours only a factor 5 is observed between vectorial and scalar runs. For the whole program, the vectorisation gain is then 6 (≈ 1 for 15,000 particles). Therefore, the code will be much more interesting when the physical model includes a lot of phenomena such as magnetic field, radiative transfer, etc. Our tests and comparisons with other works have proved that the variable resolution gives good results, particulary for the collapse, where the resolution is a crucial problem. Our adjustment of the kernel size h suppresses the instabilities due to its variation and is more fit to density distributions with steep gradients.

Appendix

We present in this appendix a scheme that will be performant on scalar processors.

Calculation of the ITNEIB and NNEIB arrays.

```
C   loop 1 is over the particles
      DO 1J = 1, N
                  L = LL(J)
                  M = MM(J)
C   loop 2 is over the cells of the block
                  DO 2MR = M - NC, M - NC
                  DO 2LR = L - NC, L + NC
                        KT = N1CEL(LR, MR)
C   if no particle in the cell, it is no necessary to make any
C       calculation
                        IF (KT.EQ.0) GO TO 2
```