```
C  loop 30 is over the particles
                    DO 30 J = N1, N2
                       K = I + 1
C  summation to calculate the density
                       DEN(J) = DEN(J) + W(J, I) + W(J, K)
   30             CONTINUE
   20          CONTINUE
   10       CONTINUE
```

NGROUP lies between 9 and 10, and the inner loop is the longest. We have unrolled by a factor 2 ($K = I + 1$) the loop 20 to optimize the calculations.

Between scalar and vectorial runs of the vectorized code we have observer a factor 11 in the CPU time, and a factor 6 between the vectorised and scalar codes. Figure 2 gives the variation of the CPU time versus the number of particles for the three cases (vectorised scheme with vectorisation ON and OFF and scalar scheme). Note that the N-body subroutine is not included in the evaluation of the total CPU time, but it is only of the order of 3% (therefore its dependence on N would have been washed out by that of the hydrodynamic code). As regards CPU time variation versus the particles number, a proportionality has been found (i.e., CPU time $\propto N$). This is due to the constant number of neighbours necessary to ensure good accuracy and to the method that only makes useful calculations. Indeed, we expect the CPU time to vary as $\propto N \cdot \langle \text{NNEIB} \rangle$, this is only an order of magnitude since the distribution of potential neighbours may depend on N even if $\langle \text{NNEIB} \rangle$ is fixed.

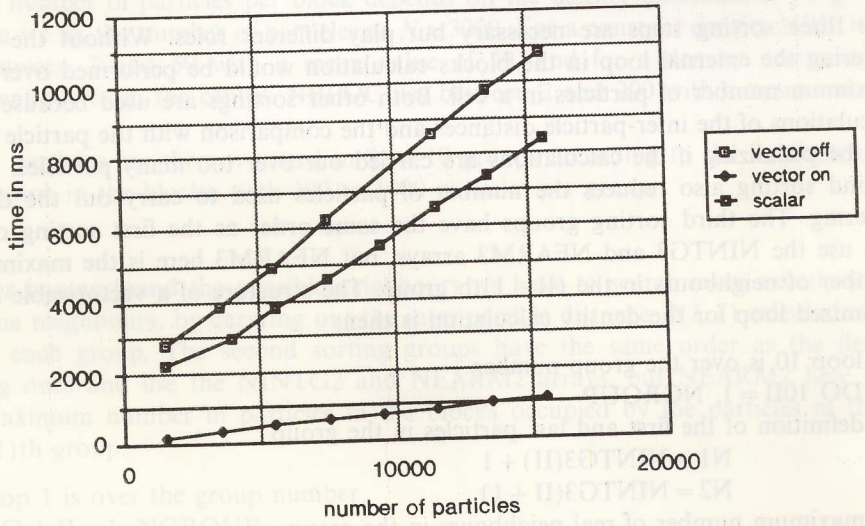

FIG. 2. CPU time per time step versus the number of particles for several schemes: vectorised scheme with vectorisation ON and OFF and scalar scheme. The calculations are performed over the 3×3, 5×5, and 7×7 blocks in the case of the collapse simulation (see Section 5). Note that the N-body subroutine is not included in the evaluation of the total CPU time, but it is only of the order of 3%.