TABLE II

Time (in ms) for Each Step of the Hydrodynamical Part of the Code

	Uniform density		$r = r_0 \exp(-x^2/a^2)$	
	1 group	optimisation	1 group	optimisation
SORT1	13	13	9	9
BLOCK	67	67	41	30
SORT2	30	23	21	11
NEIGHBOURS	66	64	43	26
SORT3	12	11	13	7
DISTANCES	34	32	41	13
DENS	12	12	13	8
VISCO	93	81	105	55
GRADIENT	2	1	1	1
Total	329	304	287	160
Ratio	1.1		1.8	
Total without sorting	274		244	
Ratio	0.9		1.5	

Note. The tests are performed over two density distributions using 3980 particles, a search over the 3×3 and 5×5 blocks and with the Gingold–Monaghan viscous tensor. For the exponential function the simulation window is $[-7, +7]$ while the half intensity width is 3.5. We give the total CPU time with and without sorting for the case where the group's number is 1, for which the sortings are useless.

group, while for steeper density gradient the gain is more important with the optimised number of groups. Moreover, the program is more efficient for the gaussian distribution.

4. Interpolation Kernels

The Kernel

Each element of fluid is described by a smoothed out distribution of density, by using an interpolation function $w(r, h)$. The construction of the interpolation kernel $w(\mathbf{r}, h)$ is guided by the requirements of accuracy, smoothness and computational efficiency [15] and then different functions have been used. Unlike the exponential function $w(r, h) = 1/8\Pi h^3 \exp(-r/h)$ chosen by Wood [19] that has nonzero derivatives in $r = 0$ and then overestimates the self-contribution of a particle, we prefer to build a kernel by

$$(\partial \mathbf{w}(r, h)/\partial r)_{r=0} = 0 \tag{4.1}$$

$$(\partial \mathbf{w}(r, h)/\partial r)_{r=h} = 0 \tag{4.2}$$

$$\mathbf{w}(r, h)_{r=h} = 0 \tag{4.3}$$