

to select its potential neighbours in the nearest cells. The number of potential neighbours is maintained at ≈ 50 by adjusting the number of surrounding cells and/or the cell size. On these potential neighbours we can then compute the inter-particle distances and determine the true neighbours. Once the neighbours have been determined for each particle, one simple method would be to sum over the same number of neighbours (i.e., $N_{\text{neighbours max}}$). But the problem is the large range in the number of neighbours, which goes from 2 to 40. This implies a lot of unnecessary calculations, namely for the particles which have a few neighbours. Therefore we will sort particles in groups, according to their number of neighbours. The details of the group dispatching are described in Section 3 (steps 1, 2, 3).

3. DETAILED DESCRIPTION OF THE ALGORITHM

To optimise the search for neighbours, a grid is superposed on the system, the cell of which is chosen to be the maximum size of a particle (h_{max}). Pre-gathering particles in groups avoids the N^2 tests on the relative distance of each couple of particles. The principal stages of the algorithm are the search for the neighbouring particles (which decomposes in a search for every particle in a given cell, chained in a linked list, and the calculation of true neighbours) and the summation over these ones (with CPU time proportional to $N_{\text{neighbours}} \times N$).

The current scalar scheme is then:

— **The linked lists.**

```

C Loop over the number of particles
  DO 1J = 1, N
    LL = L(J)
    MM = M(J)
C if first particle found, particle J can be chained in 4
  IF (NUM(LL, MM).NE.0) GO TO 4
C if first particle not found, J is the first
  NUM(LL, MM) = J
  GO TO 1
C search for the first null element in ICHAIN to place J
  4 K = NUM(LL, MM)
  6 KP = ICHAIN(K)
  IF (KP.EQ.0) GO TO 8
  K = KP
  GO TO 6
  8 ICHAIN(K) = J
  1 CONTINUE

```

where L and M are the coordinates of the cell containing the particle J , $\text{NUM}(L, M)$ the number of the first particle found in the cell (L, M) , and ICHAIN