the array of numbers of successive particles of the same chain (i.e., found in the same cell).

— **The search for the neighbours** in the nine surrounding cells including the cell of the particle and the calculation of the macroscopic quantities (here $\rho = \text{den}$):

```
           DO 23J = 1, N
    23     DEN(J) = 0
C   loop 2 is over the number of particles
           DO 2J = 1, N
                    LX = L(J) + 1
                    LN = L(J) − 1
                    MX = M(J) + 1
                    MN = M(J)
C   loop 21 is over the surrounding cells
                    DO 21 LL = LN, LX
                    DO 21 MM = MN, MX
                           KT = NUM(LL, MM)
C   If no particle in the cell it is not necessary to calculate W and DEN
                           IF (KT.EQ.0) GO TO 21
    22                     CONTINUE
C   Calculation of W(KT, J)
                           - - - - - - - - - - - - - - - -
C   calculation of the density
                           DEN(J) = DEN(J) + W(KT, J)
C   Use of the chaining to find the neighbours of J and sum
                           KP = ICHAIN(KT)
                           IF (KP.EQ.0) GO TO 21
                           KT = KP
                           GO TO 22
    21              CONTINUE
    2      CONTINUE
```

Another more time-effective scalar scheme is given in the Appendix, but this one is interesting because it needs less memory. These two loops are not vectorisable and our first goal is to suppress the GO TO instructions. The solution to vectorise lies in the calculation in advance of the neighbours of a given particle and then the second loop will be replaced by the scheme 1:

```
           DO 10J = 1, N
                DO 20 KT = 1, NNEIB(J)
                - - physical values calculation - - - - - - - - - -
    20          CONTINUE
    10     CONTINUE
```