Computational Physics

29-January-2001

Ex. 1a: Random Walk on a square lattice

In a random walk (RW) on a two-dimensional square lattice, at each step, a particle can move up, down, left or right, with equal probability. Make use of a random number generator to decide the direction of each step. In fortran 90 you can use the built in function RANDOM_NUMBER(ran) which returns values of ran uniformely distributed between 0 an 1 ($0 \leq ran < 1$). You must initialize (or restart) the random sequence by calling the function RANDOM_SEED. An example can be found in the section fortran90.

Make some test of the randomness of this generator, either before you start or during the calculation, for instance by checking how many steps have been made to the right, left, up or down. We are not interested in keeping track of the actual trajectory in space but are only interested in calculating the mean square displacement as a function of the number of steps $nstep$ for $nstep = 2, 4, 8, 16, 32, 64$ with an accuracy of 5 % or less.

Establish the dependence of the mean square displacement

$$< \Delta R^2(N) >=< x^2(N) > - < x(N) >^2 + < y^2(N) > - < y(N) >^2 \qquad (1)$$

as a function of the number of steps $nstep$ by assuming the asymptotic N dependence of $< \Delta R^2(N) >$ to be

$$< \Delta R^2(N) >\sim N^{2\nu} \qquad (N >> 1) \qquad (2)$$

If $\nu \sim 1/2$, estimate the self-diffusion coefficient D given by

$$< \Delta R^2(N) >\sim 2dDN \qquad (3)$$

with d=2 in two dimensions.

A useful exercise to test your program is to enumerate all the possible walks on a square lattice of size $N \times N$, with $N = 2, 4$ and obtain exact results for $< x(N) >$, $< y(N) >$ and $< \Delta R^2(N) >$ to compare with your numerical results.