LATTICE MODEL OF DIFFUSION IN 2D. DEPENDENCE OF DIFFUSION COEFFICIENT ON DENSITY

Bartosz Banasik

April 2, 2019

1. Below are located to graphs. 1 illustrating diffusion coefficient D for different values of the concentration C, against MCS (Moste Carlo steps) "time". 2 ilustrating the density coefficient D against concentration C. Exepriments were done for `20x20` lattice and `20` independent realization.

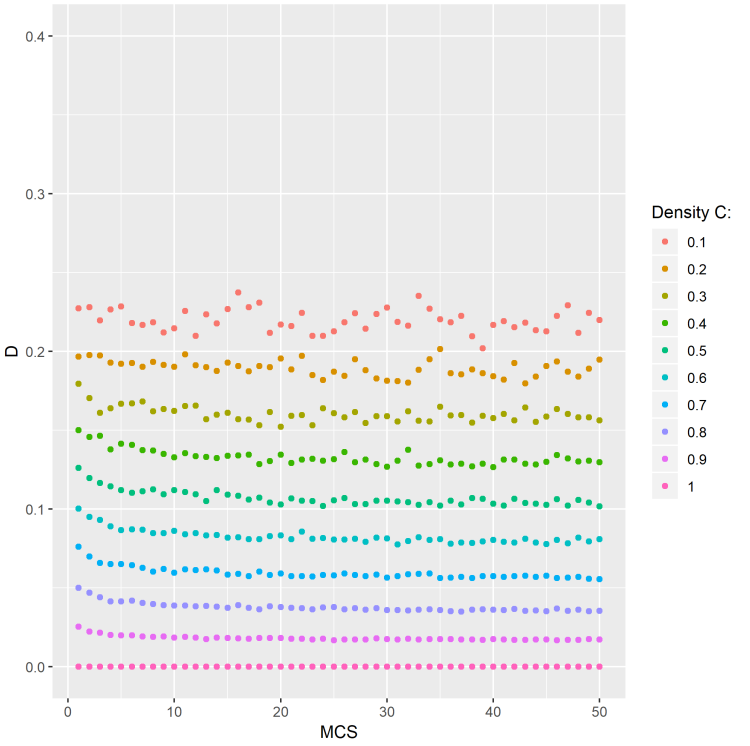

Figure 1

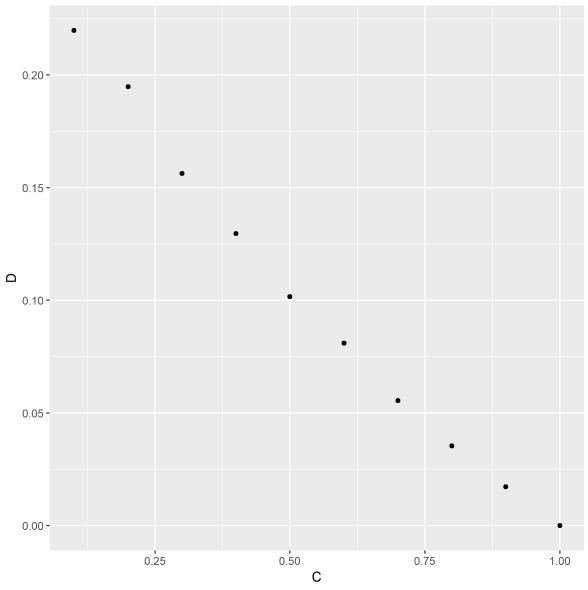

Figure 2

2. Source code for the solution.

```
implicit none
! placeholder for random calls
REAL :: random
! size of the lattice
integer, parameter :: L=20
! number of realizations
integer, parameter :: S=20
! number of max monte carlo steps
integer :: K, MCS=50
! denisty of the system
real :: C=0.0
! number of particles
integer :: N = 0
! table of nearest neighbour
integer, dimension(L) :: IN, IP
! table of occupies locations
logical, dimension(L,L) :: A
! iterator variables
integer :: i,j,ii
! actual location for all particles
integer, dimension(:), allocatable :: x, y
! temporary variables
integer :: xt, yt, dxt, dyt
! dispacement arrays
integer, dimension(:), allocatable :: dx, dy
! square dispacements
real :: dr=0, drS = 0

open(unit=2, file='results1.csv')
write(2,*) "C, MCS, D"

open(unit=3, file='results2.csv')
write(3,*) "C, D"

do i=1,L
  IN(i) = i+1
  IP(i) = i-1
enddo
IN(L) = 1
IP(1) = L

! start loop for different
! density of the gas
do C=0.1,1,0.1
  N = int(C*L**2)
  allocate(x(N), y(N))
  allocate(dx(N), dy(N))

  ! start monte carlo simulation
  ! with different values of steps
  do k=1,MCS
    drS = 0
    ! start realization
    do ii=1,S
    ! initialize dispacement arrays
    do i=1,N
      dx(i) = 0
      dy(i) = 0
    enddo

    ! initialize A array
    do i=1,L
      do j=1,L
        A(i,j) = .TRUE.
      enddo
    enddo
    dyt = 0
    call random_number(random)
    SELECT CASE (int(4*random))
      CASE (0)
          xt = IN(x(j))
          dxt = 1
      CASE (1)
          xt = IP(x(j))
          dxt = -1
      CASE (2)
          yt = IN(y(j))
          dyt = 1
      CASE (3)
          yt = IP(y(j))
          dyt = -1
    ENDSELECT
    if (A(xt,yt)) then
        A(xt, yt) = .FALSE.
        A(x(j), y(j)) = .TRUE.
        x(j) = xt
        y(j) = yt
        dx(j) = dx(j) + dxt
        dy(j) = dy(j) + dyt
    endif
  enddo
enddo
! end of the monte carlo simulation
! for a given K

! compute square displacement
dr = 0
do i=1,n
  dr = dr + dx(i)**2 + dy(i)**2
enddo
! average for 1 simulation for N particles
dr = dr/float(n)
drS = drS + dr

enddo
! end loop for many simluations

!average drS for simulations
drS = drS/float(S)
drS = drS/float(4 * K)
write(2,*) C, ',', K, ',', drS

enddo
! end of the monte carlo simulation
! for different values of steps

write(3,*) C, ',', drs
deallocate(x, y, dx, dy)
enddo
close(2)
close(3)
end
```