FILE: FRAPEXP.F90

```
1  program rw
2
3     use global_env
4     use calendar
5     use walker
6     use cell
7     use frap
8
9     implicit none
10
11    integer :: max_events
12    integer :: action, opt1, opt2
13
14    real(DP) :: time, time_limit, time_stat
15    real(DP) :: sigma, rho, xx
16
17    integer, dimension(:,:), allocatable :: p0
18
19    real(DP), dimension(:), allocatable :: rn
20
21    integer :: i, j, k, l, m, n
22    integer :: n_errors
23
24    character(len=10) :: arg
25
26 !!!!!! My Variables
27
28    logical :: bleached = .false.
29    real(DP) :: ratio
30
```

```
31      open(72, file = 'frap.dat')
32
33  !!!!!!!!!!!!!!!!!!!!!!
34
35      ! Extract the needed data from the configuration file
36
37      open(10,file='config.dat')
38      read(10,*) n_walkers
39      read(10,*) box_size
40      read(10,*) cell_size
41      read(10,*) k_3d, k_2d
42      read(10,*) k_on, k_off
43      read(10,*) time_limit
44      read(10,*) time_stat
45      close(10)
46
47      ! open log file for output
48
49      open(fmain,file='rw.log')
50
51      write(fmain,*) '*** KMC FRAP *** '
52      write(fmain,*) 'Started at ',time_stamp()
53      write(fmain,'(/a,i5)')  'Number of Random Walkers=',n_walkers
54      write(fmain,'(3(a,i4))') 'Box size=',box_size(X),' x',
           box_size(Y),' x',box_size(Y)
55      write(fmain,'(3(a,i4))') 'Cell size=',cell_size(X),' x',
           cell_size(Y),' x',cell_size(Y)
56      write(fmain,'(2(a,f8.4),a)') 'Transition rate=',k_3d,'(3d)
           ',k_2d,'(2d)'
57      write(fmain,'(2(a,f8.4),a)') 'Transition rate=',k_on,'(on)
           ',k_off,'(off)'
```

```
58    write(fmain,'(a,e15.3)') 'Total simulation time=',time_limit
59    write(fmain,*)
60
61    ! open a data file #1
62    write(fmain,*)  'outout: onoff.dat'
63    open(fdat1,file='onoff.dat')
64
65    ! open a data file #2
66    write(fmain,*) 'output: diffusion.dat'
67    open(fdat2,file='diffusion.dat')
68
69    ! open a file for debug information, iff the debug flag is
         issued
70    if(debug) then
71       write(fmain,*) 'output: debug.dat'
72       open(fdbg,file='debug.dat')
73    endif
74
75    ! set the maximum number of walkers.
76    !(this determines the size of array)
77    max_walkers = 2*n_walkers
78
79    ! set the maximum number of events recorded in the calendar
         at a time.
80    max_events = 2*max_walkers
81
82    ! initialize the event calendar
83    call calendar_init(n_walkers,max_events)
84
85    ! initialize the random walkers
86    call walker_init(n_walkers)
```

```
87
88      ! initial position of walkers
89      !(for example uniformally random distribution)
90      allocate(rn(n_walkers)) ! Create a vector for the random
          numbers
91      call random_number(rn)
92      walker_pos(1:n_walkers,X) = ceiling(rn*box_size(X))
93      call random_number(rn)
94      walker_pos(1:n_walkers,Y) = ceiling(rn*box_size(Y))
95      call random_number(rn)
96      walker_pos(1:n_walkers,Z) = ceiling(rn*box_size(Z))
97      deallocate(rn) ! Not needed anymore, so there isn't any sense
          in
98      ! letting rn take up any memory
99
100     ! initialize the cell configuration
101     call cell_init(n_walkers,max_walkers,walker_pos)
102     ! This is used for determining particle collisions
103
104     if(debug) call cell_test(n_walkers)  ! check the consistency
          in cell assignment
105
106     ! predict the jump time for all walkers
107     do i=1,n_walkers
108          call walker_predict_event(i)
109     end do
110
111     ! save the initial position
112     allocate(p0(max_walkers,3))
113     p0 = walker_pos
114
```

```
115     ! schedule the first data output
116     call calendar_schedule_event(11,0,0,time_stat)
117     call calendar_schedule_event(12,0,0,time_stat)
118
119     n_collisions = 0
120     time_current = 0.0_DP
121
122     do while (time_current<=time_limit)
123
124        ! find the next event to happen
125        call calendar_find_event(action,opt1,opt2)
126
127        ! execute the event
128        if (action<=10)  then   ! simple jump event
129
130           ! walker 'opt1' jumps
131           call walker_action(opt1,action)
132           ! predict next jump of walker 'opt1'
133           call walker_predict_event(opt1)
134
135        else if(action<=20) then   ! non-physical events
136
137           select case(action)
138              case(11)
139                 !  evaluate surface density
140                 sigma = count(walker_pos(:,Z)>box_size(Z))
141                 rho = count(walker_pos(:,Z)==box_size(Z))
142                 write(fdat1,'(f10.2,2e13.5)') time_current, rho,
                       sigma
143                 !  set the next evaluation time
144                 call calendar_schedule_event(11,0,0,time_current+
```

```
                time_stat)
145          case(12)
146             !  evaluate mean square displacement
147             xx = real(sum((walker_pos-p0)**2),kind=DP)/real(
                   n_walkers,kind=DP)
148             write(fdat2,'(f10.2,e13.5)') time_current, xx
149             !  set the next evaluation time
150             call calendar_schedule_event(12,0,0,time_current+
                   time_stat)
151          case default
152             ! unknown event
153             write(fmain,'(f10.2,a)') time_current, 'unknown
                   event'
154             ! emergency stop
155             stop
156       end select
157
158    else if(action>20) then
159       ! binary event (\such as reaction
160       write(fmain,'(f10.2,a)') time_current, 'binary event'
161       ! binary events have not implimented yet.
162    end if
163
164    !!!!!!!!!!! My FRAP Code
165
166    if (time_current .gt. time_limit / 10) then
167
168      if (bleached .eqv. .false.) then
169          call bleach()
170          bleached = .true.
171      end if
```

```
172
173           call measure(ratio)
174           write(72, *) time_current, ratio
175         end if
176
177         !!!!!!!!!!!!!!!!!!!!!!!!!!!!1
178
179       end do
180
181       close(72)
182
183       write(fmain,'(a,i10)') 'Number of collisions =',n_collisions
184
185       if(debug) call cell_test(n_walkers)
186
187       ! close output files
188
189       close(fdat1)
190       close(fdat2)
191
192       !
193       open(fdat1,file='walkers.dat')
194       ! creation of particles has not been considered yet.
195       write(fdat1,'(3i5)') (walker_pos(i,:), i=1,n_walkers)
196       close(fdat1)
197
198       if(debug) close(fdbg)
199
200       write(fmain,*) '*** Done at ',time_stamp(), '***'
201   end program rw
```