

```

1 subroutine f_main()
2   use fdp_module
3   use user_defined_types
4   implicit none
5   doubleprecision, parameter :: time_end = 10.0d0
6   doubleprecision, parameter :: dt = 1.0d0/128.0d0
7   integer :: i, j, k, ierr
8   integer :: psys_num, dinfo_num, tree_num
9   character(len=64) :: tree_type
10  doubleprecision :: time_sys = 0.0d0
11  type(fdp_controller) :: fdp_ctrl
12  call fdp_ctrl%PS_initialize()
13  call fdp_ctrl%create_dinfo(dinfo_num)
14  call fdp_ctrl%init_dinfo(dinfo_num)
15  call fdp_ctrl%create_psys(psys_num, 'full_particle')
16  call fdp_ctrl%init_psys(psys_num)
17  tree_type = "Long, full_particle, full_particle, full_particle, Monopole"
18  call fdp_ctrl%create_tree(tree_num, tree_type)
19  call fdp_ctrl%init_tree(tree_num, 0)
20  call read_IC(fdp_ctrl, psys_num)
21  call calc_gravity(fdp_ctrl, psys_num, dinfo_num, tree_num)
22  do
23    call kick(fdp_ctrl, psys_num, 0.5d0*dt)
24    time_sys = time_sys + dt
25    call drift(fdp_ctrl, psys_num, dt)
26    call calc_gravity(fdp_ctrl, psys_num, dinfo_num, tree_num)
27    call kick(fdp_ctrl, psys_num, 0.5d0*dt)
28    if (time_sys >= time_end) exit
29  enddo
30  call fdp_ctrl%PS_Finalize()
31 end subroutine f_main
32
33 subroutine calc_gravity(fdp_ctrl, psys_num, dinfo_num, tree_num)
34   use fdp_module
35   use user_defined_types
36   implicit none
37   type(fdp_controller), intent(IN) :: fdp_ctrl
38   integer, intent(IN) :: psys_num, dinfo_num, tree_num
39   type(c_funptr) :: pfunc_ep_ep, pfunc_ep_sp
40   call fdp_ctrl%decompose_domain_all(dinfo_num, psys_num)
41   call fdp_ctrl%exchange_particle(psys_num, dinfo_num)
42   pfunc_ep_ep = c_funloc(calc_gravity_pp)
43   pfunc_ep_sp = c_funloc(calc_gravity_psp)
44   call fdp_ctrl%calc_force_all_and_write_back(tree_num, &
45                                           pfunc_ep_ep, &
46                                           pfunc_ep_sp, &
47                                           psys_num, &
48                                           dinfo_num)
49 end subroutine calc_gravity
50
51 subroutine kick(fdp_ctrl, psys_num, dt)
52   use fdp_vector
53   use fdp_module
54   use user_defined_types
55   implicit none
56   type(fdp_controller), intent(IN) :: fdp_ctrl
57   integer, intent(IN) :: psys_num
58   doubleprecision, intent(IN) :: dt
59   integer :: i, nptcl_loc
60   type(full_particle), dimension(:), pointer :: ptcl
61   nptcl_loc = fdp_ctrl%get_nptcl_loc(psys_num)
62   call fdp_ctrl%get_psys_fptr(psys_num, ptcl)
63   doi = 1, nptcl_loc
64   ptcl(i)%vel = ptcl(i)%vel + ptcl(i)%acc*dt
65   enddo
66   nullify(ptcl)
67 end subroutine kick
68
69 subroutine drift(fdp_ctrl, psys_num, dt)
70   use fdp_vector
71   use fdp_module
72   use user_defined_types
73   implicit none
74   type(fdp_controller), intent(IN) :: fdp_ctrl
75   integer, intent(IN) :: psys_num
76   doubleprecision, intent(IN) :: dt
77   integer :: i, nptcl_loc
78   type(full_particle), dimension(:), pointer :: ptcl
79   nptcl_loc = fdp_ctrl%get_nptcl_loc(psys_num)
80   call fdp_ctrl%get_psys_fptr(psys_num, ptcl)
81   doi = 1, nptcl_loc
82   ptcl(i)%pos = ptcl(i)%pos + ptcl(i)%vel*dt
83   enddo
84   nullify(ptcl)
85 end subroutine drift
86
87 subroutine read_IC(fdp_ctrl, psys_num)
88   use fdp_module
89   use user_defined_types
90   implicit none
91   type(fdp_controller), intent(IN) :: fdp_ctrl
92   integer, intent(IN) :: psys_num
93   character(len=16), parameter :: root_dir = "input_data"
94   character(len=16), parameter :: file_prefix = "proc"
95   integer :: i, myrank, nptcl_loc
96   character(len=64) :: fname, proc_num
97   type(full_particle), dimension(:), pointer :: ptcl
98   myrank = fdp_ctrl%get_rank()
99   write(proc_num, "(i5.5)") myrank
100  fname = trim(root_dir) // "/" &
101        // trim(file_prefix) // "proc_num/" & ".dat"
102  open(unit=9, file=trim(fname), action='read', form='unformatted', &
103        access='stream', status='old')
104  read(9) nptcl_loc
105  call fdp_ctrl%set_nptcl_loc(psys_num, nptcl_loc)
106  call fdp_ctrl%get_psys_fptr(psys_num, ptcl)
107  doi = 1, nptcl_loc
108  read(9) ptcl(i)%id, ptcl(i)%mass, ptcl(i)%eps, &
109        ptcl(i)%pos%x, ptcl(i)%pos%y, ptcl(i)%pos%z, &
110        ptcl(i)%vel%x, ptcl(i)%vel%y, ptcl(i)%vel%z
111  enddo
112  close(unit=9)
113  nullify(ptcl)
114 end subroutine read_IC

```