

```

1  subroutine calc_gravity_pp(ep_i,n_ip,ep_j,n_jp,f) bind(c)
2      integer(c_int), intent(in), value :: n_ip,n_jp
3      type(full_particle), dimension(n_ip), intent(in) :: ep_i
4      type(full_particle), dimension(n_jp), intent(in) :: ep_j
5      type(full_particle), dimension(n_ip), intent(inout) :: f
6      integer(c_int) :: i,j
7      real(c_double) :: eps2,poti,r3_inv,r_inv
8      real(c_double), dimension(3) :: xi,ai,rij
9      real(c_double), dimension(n_jp) :: mj
10     real(c_double), dimension(3,n_jp) :: xj
11     do j=1,n_jp
12         mj(j) = ep_j(j)%mass
13         xj(1,j) = ep_j(j)%pos%x
14         xj(2,j) = ep_j(j)%pos%y
15         xj(3,j) = ep_j(j)%pos%z
16     end do
17     do i=1,n_ip
18         eps2 = ep_i(i)%eps * ep_i(i)%eps
19         xi(1) = ep_i(i)%pos%x
20         xi(2) = ep_i(i)%pos%y
21         xi(3) = ep_i(i)%pos%z
22         ai = 0.0d0
23         poti = 0.0d0
24         do j=1,n_jp
25             rij(1) = xi(1) - xj(1,j)
26             rij(2) = xi(2) - xj(2,j)
27             rij(3) = xi(3) - xj(3,j)
28             r3_inv = rij(1)*rij(1) &
29                 + rij(2)*rij(2) &
30                 + rij(3)*rij(3) &
31                 + eps2
32             r_inv = 1.0d0/sqrt(r3_inv)
33             r3_inv = r_inv * r_inv
34             r_inv = r_inv * mj(j)
35             r3_inv = r3_inv * r_inv
36             ai(1) = ai(1) - r3_inv * rij(1)
37             ai(2) = ai(2) - r3_inv * rij(2)
38             ai(3) = ai(3) - r3_inv * rij(3)
39             poti = poti - r_inv
40         end do
41         f(i)%pot = f(i)%pot + poti
42         f(i)%acc%x = f(i)%acc%x + ai(1)
43         f(i)%acc%y = f(i)%acc%y + ai(2)
44         f(i)%acc%z = f(i)%acc%z + ai(3)
45     end do
46 end subroutine calc_gravity_pp
47

```