```
 1  function accel, loc, input, magcoord, which
 2
 3  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 4  ;;
 5  ;; Computes acceleration on a packet due to the specified forces
 6  ;; Adapted from accel in rkas.pro
 7  ;;
 8  ;; Forces given in units of Rplan/s^2
 9  ;;
10  ;; Possible forces:
11  ;; * Gravity
12  ;; * radiation pressure
13  ;; * Lorentz [not yet]
14  ;;
15  ;; Equations of motion:
16  ;;   dvxdt = sum_objects (GM * (x-x_obj))/(r_obj)^3
17  ;;   dvydt = sum_objects (GM * (y-y_obj))/(r_obj)^3 + C* gamma(vy)
18  ;;   dvzdt = sum_objects (GM * (z-z_obj))/(r_obj)^3
19  ;;   -- r_obj = sqrt( (x-x_obj)^2 + (y-y_obj)^2 + (z-z_obj)^2 )
20  ;;   -- radiation pressure only valid for Na
21  ;;   dndt = instantaneous change in density
22  ;;
23  ;; Version history:
24  ;;   3.0: 7/21/2010
25  ;;       * rewritten with new structure architecture
26  ;;   2.1: 4/29/10
27  ;;       *
28  ;;       * adding optional out_of_shadow input
29  ;;   2.0: Rewritten so that individual forces are farmed out
30  ;;   1.0: Original version
31  ;;
32  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
33
34  agrav = (input.Forces.gravity) $
35    ? gravity(loc, input.geometry, input.options, which) $
36    : 0.
37
38  aradpres = (input.Forces.radpres) $
39    ? radiation_pressure(loc, input.geometry, input.options.atom, *magcoord.out_of_shadow) $
40    : 0.
41
42  alor = (input.Forces.Lorentz) $
43    ? Lorentz(loc, options) $
44    : 0.
45
46  accel = {dvdt:ptr_new(0)}
47  *accel.dvdt = agrav + aradpres + alor
48
49  q = where(finite(*accel.dvdt) EQ 0) & if (q[0] NE -1) then stop
50  return, accel
51  end
```