

```

1  pro exosphere_distribution, input, output, npack, seed
2
3  //////////////////////////////////////
4  ;;
5  ;; Distribute packets from a spherically symmetric exosphere
6  ;;   f(r) = r^b
7  ;;   or
8  ;;   f(r) = exp(-r/h)
9  ;;
10 ;; Version History
11 ;; 2.1: 20 November 2009
12 ;;   * Added option to prevent packet creation in planet's geometric shadow
13 ;;   * Added option to choose between specifying a scale height or a powerlaw exponent
14 ;; 2.0: File created
15 ;;
16 //////////////////////////////////////
17
18 stop
19 todo = lindgen(npack)
20 SpatialDist = input.SpatialDist
21
22 ;; Set the angular distribution
23 ll = !dpi*dindgen(1001)/1000. - !dpi/2.
24 f_lat = cos(ll)
25
26 r = findgen(10001)/100.+1
27 r = r[where(r LE SpatialDist.rmax)]
28 case (SpatialDist.exotype) of
29 'powerlaw': f_r = r^SpatialDist.b
30 'exponential': f_r = exp(-(r-1)/SpatialDist.b)
31 endcase
32 f_r[0] = 0. ;; Don't allow packets to start right at the surface
33
34 *output.x0 = dblarr(npack)
35 *output.y0 = dblarr(npack)
36 *output.z0 = dblarr(npack)
37 while (npack GT 0) do begin
38   lat = MonteCarloDistribution(ll, f_lat, npack)
39   lon = 2*!dpi * random_nr(seed=seed, npack)
40
41   rr = MonteCarloDistribution(r, f_r, npack)
42   q = where(rr LT 1. or rr GT SpatialDist.rmax, nq)
43   while (nq NE 0) do begin
44     w = MonteCarloDistribution(r, f_r, nq)
45     rr[q] = w
46     q = where(rr LT 1. or rr GT SpatialDist.rmax, nq)
47   endwhile
48
49   if strcmp(input.geometry.planet, input.geometry.StartPoint, /fold) then begin
50     ;; Starting at a planet
51     (*output.x0)[todo] = double(rr * sin(lon)*cos(lat)) ;; longitude = 0 => -yaxis

```

```

52 (*output.y0)[todo] = -double(rr * cos(lon)*cos(lat)) ;; longitude = 90 => axis
53 (*output.z0)[todo] = double(rr * sin(lat))
54 endif else begin
55 ;; Starting at a satellite
56 (*output.x0)[todo] = -double(rr * sin(lon)*cos(lat)) ;; longitude = 0 => -yaxis
57 (*output.y0)[todo] = -double(rr * cos(lon)*cos(lat)) ;; longitude = 90 => axis
58 (*output.z0)[todo] = double(rr * sin(lat))
59 endelse
60
61 rho = *output.x0^2 + *output.z0^2
62
63 ;; not working right if block_shadow and starting at a satellite
64 if (spatialdist.block_shadow) and ~(strcmp(input.geometry.planet, $
65 input.geometry.StartPoint, /fold)) then stop
66
67 if (spatialdist.block_shadow) $
68 then todo = where((rho LE 1) and (*output.y0 GT 0), npack) $
69 else npack = 0
70 endwhile
71
72 end
73

```